Hindawi
International Journal of Digital Multimedia Broadcasting
Volume 2019, Article ID 5284968, 7 pages
https://doi.org/10.1155/2019/5284968

Research Article

Dynamic Soft Real-Time Scheduling with Preemption Threshold for Streaming Media

Wenle Wang ,[1] Yuan Wang,[1] JiangYan Dai ,[2] and Zhonghua Cao[3]

[1] *School of Software, Jiangxi Normal University, Nanchang 330022, Jiangxi, China*
[2] *School of Computer Engineering, Weifang University, Weifang 261061, Shandong, China*
[3] *School of Software & Communication Engineering, Jiangxi University of Finance and Economics, Nanchang 330038, Jiangxi, China*

Correspondence should be addressed to JiangYan Dai; daijyan@163.com

Received 21 September 2018; Revised 19 November 2018; Accepted 3 December 2018; Published 1 January 2019

Guest Editor: Kai Wang

Over the last decades, the advancements in networking technology and new multimedia devices have motivated the research on efficient video streaming mechanisms under wireless. We consider combing soft real-time video streaming scheduling with threshold to minimize the ineffective preemption. Based on the value density and urgency of soft real-time task, the dynamic scheduling with preemption threshold strategy (DSPT) is proposed in the paper. By analyzing the response time and preemption relationship of tasks, the preemption thresholds are assigned. Simulation results show that the DSPT strategy achieves improvements about success rate, delay time, and benefit of the system.

1. Introduction

With the advancements in networking technology and new multimedia devices, video streaming mechanisms under various network and devices have attracted more and more attention. Nowadays, multimedia communications over the internet especially wireless network are the necessarily part of modern life. Many researchers have studied on video streaming management over wireless, including video coding [1–4], network coding [5–8], and wireless communication [9–11]. In wireless networks, video streaming applications are almost the largest consumer of mobile wireless data, so scheduling video streaming in wireless networks is very important.

Video streaming scheduling in wireless networks has been developed for recent years, such as given quality on service mechanisms [12–16] and real-time adaptive [17–19]. Many of video streaming scheduling studies are about hard real-time [20–22] that requires all the packets complete before the deadline; otherwise the packets will be considered invalid and dropped. However, for some actual video streaming media application, the packets of the transmission with delay can still bring a certain service and these packets can be forwarded and represented by soft real-time tasks. The soft real-time task model with value function is also applicable to video streaming with delay [23, 24]. For soft real-time scheduling, task executing under delay-constrained can bring goodput and benefit.

Generally, real-time scheduling can be divided into preemptive scheduling and non-preemptive scheduling according to whether the task can preempt each other or not. In a dynamic network, the preemptive method is more applicable. However, arbitrary preemption between tasks will affect the scheduling performance. Therefore, considering the drawbacks of the two methods, a compromise approach is utilized to limit the arbitrary or invalid preemption by setting preemption threshold. Furthermore, the limited preemption scheduling strategy is a correct choice for improving the performance.

Soft real-time task model is conducted for the video streaming under wireless. We propose a soft real-time threshold preemption scheduling strategy by combining value density and urgency for video streaming in the paper. Simulation results show that the strategy can improve the goodput, reduce the delay, and increase the benefit comparing with the EDF and LSF methods.

In this paper, we only consider the CPU in the system scheduling and ignore the time caused by context switching. We study the video streaming transmission problem of gateway devices in wireless network at the specific level of packets of transport tiers.

Organization. The structure of the paper is organized as follows: Section 2 describes the related work. Section 3 designs the structure of DSPT strategy. Section 4 defines the task model and problem analysis in detail. Section 5 is priority and preemption thresholds description. The details of scheduling based on preemption threshold are introduced in Section 6. The simulation setup and results are given in Section 7. Finally, the conclusions and future work are discussed.

2. Related Work

In wireless networks, the scheduling strategy should ensure high quality. For this purpose, lots of studies were proposed. The works in [1, 2] enabled high-quality HTTP streaming, in which video pre-encoded into segments that can only start to play if all the packet have received. However, much packet lost for video application in the above work. The literature [3] optimized the dynamic HTTP live streaming service by segment adaption. Some literatures proposed video streaming scheduling using video coding with network coding. For example, the literatures [5, 6] optimized video streaming by combining video coding and network coding for P2P network and distributed network, respectively. However, it might suffer from bandwidth inefficiencies because of unrecoverable packets. The work [7] proposed streaming scheduling for multiple server by Markov decision processes, but it had high computational complexity. In [9], the authors designed a crosslayer optimization scheme for dynamic wireless video streaming. However, these methods cannot contain the packet urgency for streaming control.

With the emergence of various networks, the research on heterogeneous wireless networks had been motivated. The literatures [11, 19] studied the concurrent multipath transfer of the capacity-limited heterogeneous wireless platform. Another work [17] proposed a real-time adaptive video streaming deliver method over multiple wireless networks. In [23, 24], delay-constrained video transmission with multiple interface in heterogeneous network was proposed.

Some researchers considered video streaming as hard real-time; that is, the streaming packet missing the deadline will be considered invalid. For example, the literature [20] proposed a feedback-based control approach under hard real-time workload to improve admission rate. In literatures [18, 21], the author studied in hard real-time video stream allocation to improve the utilization of video decoding. Nevertheless, most video streaming data meets the characteristics of soft real-time task, so delay-constrained of video is considered acceptable in applications.

3. DSPT Detail Design

We consider video streaming deliver over wireless networks as soft real-time scheduling. The streaming packet is expressed as task, and its quality and urgency are the metrics in scheduling. The task's delay will affect the quality. Therefore, we introduced the value to measure the quality. Response time analysis (RTA) is used to test the schedulability of real-time task, and the preemption threshold-based scheduling approach is to reduce invalid preemption among tasks. The scheduling algorithm can effectively reduce the delay and improve the success rate and benefit. The flow diagram of DSPT strategy is shown in Figure 1. In this paper, we assume that the value function $V_i(t)$ of the soft real-time task T_i keeps unchanged in the interval $[a_i, d_i)$ and decreases linearly in the interval. While executing in the interval $[d_i, cr_i)$, the task T_i is not allowed to be preempted in order to protect the execution of the task.

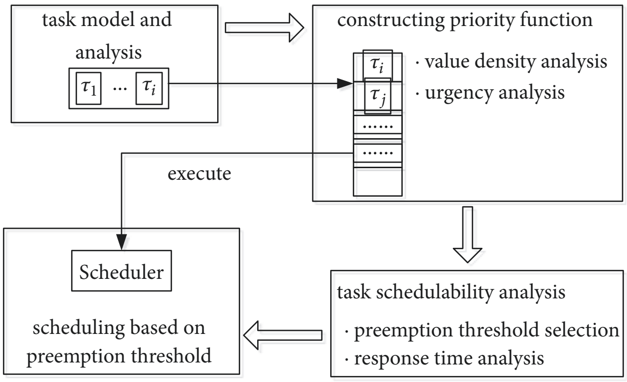


Figure 1: The flow diagram of DSPT strategy.

Contribution 1. We construct the priority assignment function combining value density and urgency, improving the success and quality of task soft real-time scheduling.

Contribution 2. We propose the task scheduling strategy DSPT based on preemption threshold, which can obtain better performance.

4. Task Model and Analysis

Considering video streaming with delay-constrained under wireless, we conduct soft real-time task model in this section.

4.1. Soft Real-Time Task Model. We refer to a task as an aperiodic task model, and the task is soft real-time. This section provides the definitions of soft real-time task model used in the paper.

Task T_i: An aperiodic task T_i is defined as a triple $T_i := \{a_i, et_i, d_i, cr_i\}$, where a_i,et_i,d_i, and cr_i are the arrival time, the estimated execution time, the relative deadline, and the critical time of task T_i.

Having executed time ht_i: ht_i denotes that the task has been executing ht_i units at time t_0.

Required execution time rt_i: rt_i denotes the remaining execution time of task T_i. Thus, it meets $rt_i = et_i - ht_i$.

Executable time lt_i: lt_i is the executable time of the task T_i prior to its deadline d_i, where at the time $t_0 (t_0 < d_i)$.

Soft Executable time slt_i: When missed deadline d_i, task can continue to execute until cr_i. slt_i is the executable time prior to the critical time cr_i, that is, $slt_i = cr_i - t_0$.

Slack time st_i: The slack time st_i denotes that the time unit after finish execution until deadline d_i, thus, $st_i = lt_i - rt_i$. st_i remains stable when the task T_i is in execution, and st_i decreases when the task T_i is preempted. Once $st_i < 0$, the task T_i cannot finish before d_i.

Delay Time dt_i: The delay time dt_i denotes the finish time of task T_i after d_i. At the time $t_0 (d_i < t_0 \leq cr_i)$, it meets $dt_i = t_0 - d_i$.

Priority pr_i: pr_i is defined as the priority of task T_i. The larger pr_i's value, the higher T_i's priority.

Threshold θ_i: θ_i is the preemption threshold of task T_i, where $\theta_i \geq pr_i$.

Value function $V_i(t)$: $V_i(t)$ defines the value of the system for executing T_i at time t, where $t \in [a_i, cr_i)$. The task T_i 's value will change when T_i keeps executing, and the function $V_i(t)$ is dynamic. $V_i(t)$ will be analyzed in detail in Section 4.

Value density function $VD_i(t)$: $VD_i(t)$ is the ratio between $V_i(t)$ and required execution time rt_i, that is, $VD_i(t) = V_i(t)/rt_i$. Obviously, $VD_i(t)$ is growing with the decreasing of rt_i.

4.2. Value and Value Density Function of Soft Real-Time Task. According to assumption, the value function $V_i(t)$ of the soft real-time task T_i at time t is defined as

$$V_i(t) = \begin{cases} V_i & t < d_i \\ \dfrac{cr_i - t}{cr_i - d_i} \times V_i & d_i \leq t < cr_i \end{cases} \tag{1}$$

From (1), we can see that when $t < d_i$, $V_i(t)$ is stable at V_i; however when $d_i \leq t < cr_i$, $V_i(t)$ is linearly diminishing until 0.

Value density function $VD_i(t)$ is denoted as

$$VD_i(t) = \begin{cases} \dfrac{V_i}{rt_i} & t < d_i \\ \dfrac{cr_i - t}{cr_i - d_i} \times \dfrac{V_i}{rt_i} & d_i \leq t < cr_i \end{cases} \tag{2}$$

We can see that $VD_i(t)$ is not only concerned with $V_i(t)$ but also related to rt_i. According to the definition, $rt_i = et_i - ht_i$. As we known, when the task T_i is executing, rt_i decreases with the growing of the time t. And when the task T_i is blocked, the rt_i remains unchanged. The discussion of $VD_i(t)$ is as follows:

(1) $t < d_i$. Within this interval, $VD_i(t) = V_i/rt_i$. When T_i is executing, $VD_i(t)$ grows with the decreasing of rt_i. While T_i is blocked, $VD_i(t)$ remains unchanged.

(2) $d_i \leq t < cr_i$. In this interval, $VD_i(t)$ can be derived as follows: $VD_i(t) = (V_i/rt_i) \times ((cr_i - t)/(cr_i - d_i))$

From the above expression, the affect part of the $VD_i(t)$ is

$$\frac{cr_i - t}{rt_i} \tag{3}$$

Because the task in scheduler can finish without any interference of any other task, $(cr_i - t)/rt_i > 1$. In this case, if the task T_i keeps executing, the molecular part and the denominator part of (3) decrease at the same time. Further analysis is needed:

(1) At t_0 time, (3) is recorded as: $(cr_i - t_0)/rt_i$. When task T_i has executed Δt from t_0, (3) is formulated to $(cr_i - (t_0 + \Delta t))/(rt_i - \Delta t)$. Because $slt_i/rt_i > 1$, then $(cr_i - (t_0 + \Delta t))/(rt_i - \Delta t) > (cr_i - t_0)/rt_i$, where $\Delta t > 0$. That is, when task T_i is executing, $VD_i(t)$ grows with the decreasing of rt_i.

(2) When the task T_i is blocked, the molecular part of (3) decreases with the growth of t, while the denominator rt_i is kept unchanged. Thus, (3) decreases with the growth of t. Therefore, Theorem 1 can be obtained.

Theorem 1. *While the task keeps executing in the progress, it increases with time t.*

4.3. Analysis of the Task's Urgency. The more urgency of the task is, the sooner execution is required.

Definition 2. At any t, the urgency $Ug_i(t)$ of task T_i is defined as $Ug_i(t) = 1/e^{st_i}$.

$Ug_i(t)$ is affected by st_i, the discussion is as follows:

(1) When T_i remains in execution, st_i does not change with the growing of the time t and then $Ug_i(t)$ remains unchanged.

(2) When T_i is blocked, st_i decreases with the growth of t, which makes $Ug_i(t)$ increase.

5. Priority and Preemption Thresholds for Soft Real-Time Tasks

Task scheduling is driven by priority, while task priority function combines value density with urgency.

5.1. The Construction of Task's Priority Function. Considering the urgency and the benefits of task, task scheduling is based on priority. Only the completed task can gain value, so relative to the value density, the urgency is more important. In system scheduling, the first consideration is system success rate and then is value brought by tasks. It means that urgency is given much greater weight than value density in priority function. Therefore, we use exponential weighting to the urgency while using logarithmic weighting to the value density. The function of priority assignment can be rewritten to

$$pr_i = Ug_i(t) \times \ln[1 + VD_i(t)] \tag{4}$$

Based on the above analysis of value density $VD_i(t)$ and urgency $Ug_i(t)$, we can conclude that

(1) When T_i executes at interval $[a_i, d_i)$, pr_i satisfies

$$pr_i = \frac{1}{e(st_i)} \times \ln\left[1 + \frac{V_i}{rt_i}\right] \tag{5}$$

Further analysis is needed:

(1) According to Theorem 1, when T_i keeps running, $\ln[1+V_i/rt_i]$ increases. Because st_i remain unchanged with time t growing. Therefore, (5) keeps increasing.

(2) Once T_i is preempted, $1/e^{st_i}$ augments as time t grows, while $\ln[1 + V_i/rt_i]$ remain unchanged, then (5) keeps increasing. Based on above, pr_i increases as time t at interval $[a_i, d_i)$.

(2) When T_i executes at $[d_i, cr_i)$, its slack time is slt_i and satisfies

$$pr_i = \frac{1}{e^{st_i}} \times \ln\left[1 + \frac{cr_i - t}{cr_i - d} \times \frac{V_i}{rt_i}\right] \quad (6)$$

(1) In the process of running T_i, with the growing of the time t, $((cr_i - t)/(cr_i - d_i)) \times (V_i/rt_i)$ increases while slt_i remains unchanged and then (6) keeps increasing.

(2) When T_i has been preempted, with the growing of time t, slt_i decreases and $1/e^{slt_i}$ augments accordingly while $((cr_i - t)/(cr_i - d_i)) \times (V_i/rt_i)$ reduces. Further discussion of Eq. (6) is needed. At moment t, $slt_i = cr_i - t - rt_i < cr_i - d_i$. After task T_i has been executed with Δt, there is $pr_i(\Delta t) = (1/e^{cr_i - t - rt_i - \Delta t}) \times \ln[1 + ((cr_i - t - \Delta t)/(cr_i - d_i)) \times (V_i/rt_i)]$, where $cr_i - t - rt_i - \Delta t > 0$. And $1/e^{cr_i - t - rt_i - \Delta t}$ is exponential growth, in which growth rate is faster than $\ln[1 + ((cr_i - t - \Delta t)/(cr_i - d_i)) \times (V_i/rt_i)]$'s logarithmic descent rate significantly. Therefore, $pr_i(\Delta t)$ increases when Δt grows.

5.2. Preemption Threshold Selection for Task. The selection of preemption thresholds directly affects the scheduling algorithm. We select preemption threshold by analyzing the task's response time.

Definition 3. Task T_i's Effect Job Set is denoted as EJS_i. During the execution of task T_i, all task of the sets can preempt T_i, which are defined as Effect Job Set (EJS_i), satisfying $EJS_i = T_m \mid (pri_m > pri_i \wedge (D_m > A_i) \wedge (A_m < D_i)$.

Response time Rt_i of task T_i contains two parts: estimated execution time et_i and sum of the remaining execution time of the effect job sets EJS_i:

$$Rt_i = et_i + \sum_{T_j \in EJS_i} rt_j \quad (7)$$

By analyzing the response time of T_i, its threshold θ_i can be obtained under the condition which can be scheduled. The procedure of preemption threshold selection is shown in Algorithm 1, where T_{max} is the highest priority task.

Obviously, the computational complexity of Algorithm 1 is $O(n)$.

```
Calculate $\theta_i$ using (7);
$\theta_i = pr_i$;
WHILE ($Rt_i \leq cr_i$) DO
  IF ($\theta_i > pr_{max}$)
     return not scheduled;
  ELSE
    IF $T_i$ is executing and $st_i < st_{max}$
       $\theta_i := pr_{max}$;
       return $\theta_i$;
    ELSE
       Calculate $Rt_i$ using (7)
    END IF
  END IF
END WHILE
return $\theta_i$
```

Algorithm 1: Compute θ_i.

6. Task Scheduling Strategy Based on Preemption Threshold

We propose a limited preemption scheduling strategy DSPT based on preemption threshold, which reduces the blocking time of the tasks by restricting preemption.

Definition 4. Sufficient and necessary conditions of $pr_i > \theta_j$ is that task T_i can preempt task T_j.

By Definition 4, if task T_i preempts task T_j, there must be $pr_i > pr_j$.

Theorem 5. *If task T_i and task T_j are not preempted, the following should be satisfied: $(pr_i \leq \theta_j) \wedge (pr_j \leq \theta_i)$.*

Proof. Supposing that there are two tasks, T_p and T_q, which do not preempt each other, $\neg((pr_p \leq \theta_q) \wedge (((pr_q \leq \theta_p)))) \Longrightarrow (pr_p > \theta_q) \vee (pr_q > \theta_p)$ is satisfied.

There are two possible cases:

(1) If $pr_p > \theta_q$, task T_p can preempt task T_q,

(2) If $pr_q > \theta_p$, task T_p may be preempted by task T_q.

Obviously, tasks T_p and T_q can preempt each other, in contradiction with Definition 4. Therefore, Theorem 5 can be proved. □

6.1. Scheduling Algorithm Based On Preemption Threshold. The soft real-time dynamic task scheduling based on preemption threshold (DSPT) called Algorithm 1 is to calculate the threshold and determines the scheduling queue dynamically as shown in Algorithm 2.

6.2. Algorithm Complexity. There is only one loop in Algorithm 2, that is, the 7th step is the loop of the length that is equal to the number of tasks in the ready queue. Thus, the computational complexity of the Algorithm 2 is $O(n \times m)$, and the computational complexity of the whole scheduling algorithm is $O(n \times m)$.

```
$\theta_i$: preemption threshold of $T_i$;
$pr_i$: priority of $T_i$;
$S_j$: ready tasks queue. $1 \le j \le num$,
$num$ is the number of tasks in the queue;
$T_{arr}$: task just arriving;
$T_{exe}$: task in executing;
$T_{max}$: the highest priority task of $S_j$;
$T_{arr}$ arrive;
IF $num = 0$;
    $T_{exe} := T_{arr}$;
    EXECUTE $T_exe$;
ELSE
    Compute $pr_{arr}$ using (7);
    FOR $i = 1$ to $num$ DO
        Compute $pr_i$ using (5);
    Compute $\theta_i$;
    SORT task in $S_j$ by $pr_j$ Desc;
END IF
IF $pr_{arr} > \theta_{max}$
    $T_{exe} := T_{arr}$;
    IF $T_{in}$ in $S_j$ can't Finish
        ABORT $T_{in}$;
    ELSE
        $T_{exe} := T_{max}$;
        ADD $T_{arr}$ to $S_j$;
        $num + +$;
    END IF
    EXECUTE $T_{exe}$;
END IF
```

Algorithm 2: Task scheduling.

7. Simulations and Analysis

7.1. Simulation Experiments. The experiment platform's CPU is Intel dual-core 2.8 GHZ processor, and the memory is 8 GB. The operating system is Ubuntu Linux, and the source code is implemented by C language.

In the experiment, we consider the aperiodic soft real-time task. Any task T_i's arrival time a_i obeys Poisson distribution ($\lambda = 4$) and the estimated execution time et_i is subject to $[3.0, 5.0]$ average distribution. Therefore, its relative deadline d_i is $[1.2, 1.5] \times et_i$ average distribution, its critical time cr_i is $[1.2, 1.5] \times d_i$, and its value V_i is randomly selected between $[10, 50]$.

Performance indicators are included as follows:

(1) Systematic success rate (SSR):

$$SSR = \frac{N_s}{N} \tag{8}$$

where N_s denotes as the number of successful tasks and N represents the number of system overages.

(2) Average task latency rate ATDR:

$$ATDR = avg\left(\sum_{T_i \in TS} \frac{el_i}{et_i}\right) \tag{9}$$

where TS denotes the successful task set and el_i and et_i denote execution latency and execution time of the success task T_i.

(3) Effective utilization Rate (EUR):

$$EUR = \frac{CT_s}{CT} \tag{10}$$

where CT_s denotes as the CPU time for successful tasks and CT represents the total CPU time in scheduling.

(4) Cumulative Value (CV):

$$CV = \sum_{T_i \in TS} V_i \tag{11}$$

where TS denotes the successful task set and represents the value of the task T_i.

7.2. Performance Analysis. Under soft real-time environment, the preemptive threshold scheduling strategy is compared with Earliest Deadline First (EDF) method and Least Slack First (LSF) method, and the simulation results and performance are analyzed as follows.

(1) Figure 2 shows simulation results for SSR scheduled by DSPT, EDF and LSF. The task's arrival interval is represented on the horizontal axis, and the vertical axis is SSR. When the arrival interval of the task is very short, the preemption among the tasks causes the SSR to be low. With the increasing of arrival interval, the preemption among tasks decreases and the SSR arguments. From Figure 2, SSR is sorted in descending order followed by DSPT, LSF, and EDF. This is due to the EDF algorithm chooses task with earliest deadline but not most urgent, which can lead to many tasks fail. While the LSF algorithm always chose the task with the shortest slack time dynamically, its unrestricted preemption causes the thrashing which can make lots of task fail. The DSPT can adapt to the system environment dynamically, and its preemption threshold can avoid the invalid preemption among tasks. (2) The simulation results of ATDR are shown in Figure 3, in which the ATDR is on the vertical axis and arrival interval is on the horizontal axis. It can be seen from Figure 3, the ATDR decreases since the arrival interval grows from 2.5, while ATDR is lower because the intervals among tasks are so short that many of these are dropped to 1.0. Moreover, EDF algorithm leads to the highest ATDR, while the DSPT method produces the lowest ATDR. Because the EDF algorithm always chooses the task with earliest deadline of slack time to execute, many tasks miss the deadline and delay. Relative to the EDF method, the LSF algorithm assigns the most urgency task to execute in time. But its unconstrained preemption may cause serious thrashing among tasks under system overload and this increases the ATDR. The DSPT approach chooses the most urgency task to execute, and limits the preemption among the tasks by the threshold, thereby reduces the delay of the task effectively.

(3) The simulation results for EUR scheduled by EDF, LSF, and DSPT are shown in Figure 4, where the horizontal

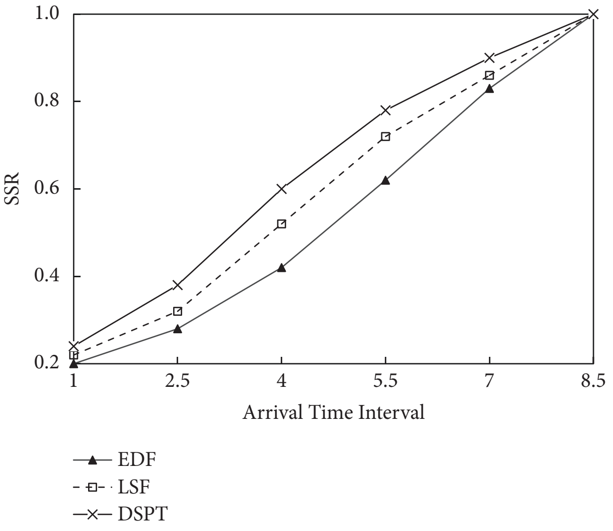


Figure 2: SSR results.

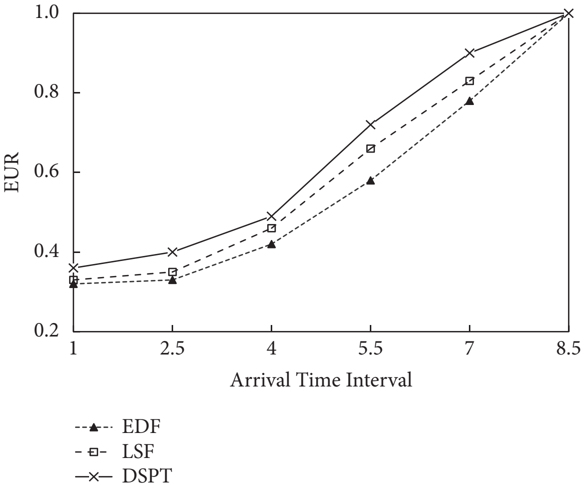


Figure 4: EUR results.

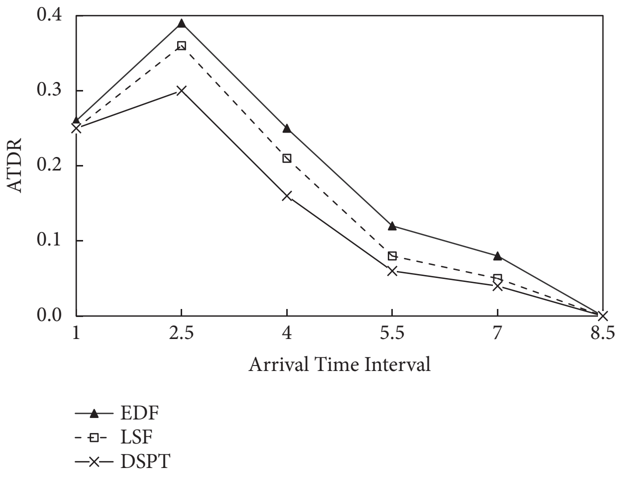


Figure 3: ATDR results.

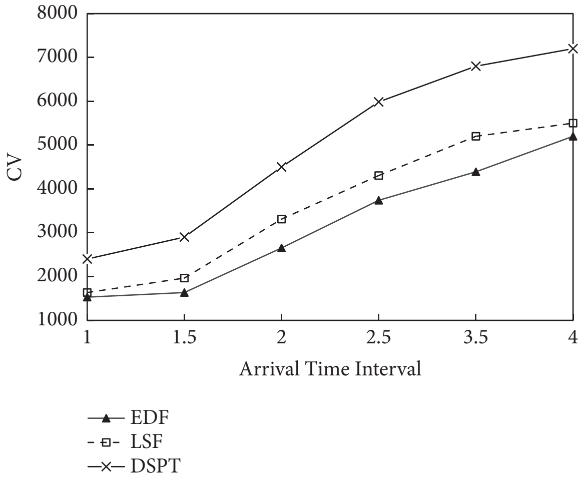


Figure 5: CV results.

axis is task's arrival interval and the vertical axis is EUR. The short arrival interval means intense preemption among tasks, which causes the EUR low. While the arrival interval becomes longer, there are EUR arguments as the tasks' preemption decreases. From Figure 4, DSPT obtains the best EUR and EDF has the least EUR. Because the EDF algorithm always execute the task with earliest deadline, which causes more invalid CPU utilization than LSF along with the task failure. The DSPT using preemption threshold can improve the EUR through reducing invalid preemption of tasks.

(4) Figure 5 plots the system's CV as arrival interval grows. It can be seen from Figure 5, the CV increases when the arrival interval of tasks increases. DSPT obtains the highest CV, because it assigns higher executing priority of the task with high value density. The LSF method produces better CV than EDF method, because under the same system load, that more tasks finish augments CV obtained with the formal approach.

8. Conclusions and Future Works

In this paper, we consider video streaming scheduling in wireless networks as soft real-time scheduling. The soft real-time task model has been constructed to express streaming packet and its quality, measure as value, and urgency metric are explored in scheduling. Response time analysis (RTA) approach has been used to test the task's schedulability, and preemption threshold has been added to the schedule to release the invalid preemption. The scheduling schema DSPT can constrain the delay and improve the success ratio and benefit effectively.

In future works, we want to apply our scheduling method to practical wireless networks, like 5G net, to test its practical effect. Moreover, the stream scheduling of mixed hard real-time and soft real-time characterizes are also a study work in the next step.

Data Availability

The data, including task's properties and performance indicators in the experiments, used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (NSFC) (Grant nos. 61562044, 41661083) and the Science and Technology Research Project of Jiangxi Provincial Department of Education (Grant nos. GJJ170234, GJJ160781).

References

[1] L. De Cicco and S. Mascolo, "An adaptive video streaming control system: Modeling, validation, and performance evaluation," *IEEE/ACM Transactions on Networking*, vol. 22, no. 2, pp. 526–539, 2014.

[2] O. Oyman and S. Singh, "Quality of experience for HTTP adaptive streaming services," *IEEE Communications Magazine*, vol. 50, no. 4, pp. 20–27, 2012.

[3] T. C. Thang, H. T. Le, A. T. Pham, and Y. M. Ro, "An evaluation of bitrate adaptation methods for HTTP live streaming," *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 4, pp. 693–705, 2014.

[4] L. He and F. Li, "Content and buffer status aware packet scheduling and resource management framework for video streaming over LTE system," *Eurasip Journal on Image and Video Processing*, vol. 2017, no. 1, 2017.

[5] A. M. Sheikh, A. Fiandrotti, and E. Magli, "Distributed scheduling for scalable P2P video streaming with network coding," in *Proceedings of the 32nd IEEE Conference on Computer Communications (IEEE INFOCOM '13)*, pp. 11-12, April 2013.

[6] A. M. Sheikh, A. Fiandrotti, and E. Magli, "Distributed scheduling for low-delay and loss-resilient media streaming with network coding," *IEEE Transactions on Multimedia*, vol. 16, no. 8, pp. 2294–2306, 2014.

[7] N. Thomos, E. Kurdoglu, P. Frossard, and M. Van Der Schaar, "Adaptive prioritized random linear coding and scheduling for layered data delivery from multiple servers," *IEEE Transactions on Multimedia*, vol. 17, no. 6, pp. 893–906, 2015.

[8] S. Huang, E. Izquierdo, and P. Hao, "Adaptive packet scheduling for scalable video streaming with network coding," *Journal of Visual Communication and Image Representation*, vol. 43, pp. 10–20, 2017.

[9] M. Zhao, X. Gong, J. Liang, W. Wang, X. Que, and S. Cheng, "QoE-driven cross-layer optimization for wireless dynamic adaptive streaming of scalable videos over HTTP," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 3, pp. 451–465, 2015.

[10] D.-R. Chen, "A real-time streaming control for quality-of-service coexisting wireless body area networks," *Applied Soft Computing*, 2017.

[11] Y. Cao, Q. Liu, Y. Zuo, G. Luo, H. Wang, and M. Huang, "Receiver-assisted cellular/wifi handover management for efficient multipath multimedia delivery in heterogeneous wireless networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2016, no. 1, 2016.

[12] C. Sieber, T. Hosfeld, T. Zinner, P. Tran-Gia, and C. Timmerer, "Implementation and user-centric comparison of a novel adaptation logic for DASH with SVC," in *Proceedings of the 2013 IFIP/IEEE International Symposium on Integrated Network Management, IM 2013*, pp. 1318–1323, Belgium, May 2013.

[13] S. A. Hosseini, F. Fund, and S. S. Panwar, "(Not) yet another policy for scalable video delivery to mobile users," in *Proceedings of the 7th ACM Workshop on Mobile Video, MoVid 2015*, pp. 17–22, USA, March 2015.

[14] S. A. Hosseini and S. S. Panwar, "Restless streaming bandits: Scheduling scalable video in wireless networks," in *Proceedings of the 2017 55th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 620–628, Monticello, IL, USA, October 2017.

[15] Y.-H. Yeh, Y.-C. Lai, Y.-H. Chen, and C.-N. Lai, "A bandwidth allocation algorithm with channel quality and QoS aware for IEEE 802.16 base stations," *International Journal of Communication Systems*, vol. 27, no. 10, pp. 1601–1615, 2014.

[16] J. Kim and E.-S. Ryu, "QoS optimal real-time video streaming in distributed wireless image-sensing platforms," *Journal of Real-Time Image Processing*, vol. 13, no. 3, pp. 547–556, 2017.

[17] M. Xing, S. Xiang, and L. Cai, "A real-time adaptive algorithm for video streaming over multiple wireless access networks," *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 4, pp. 795–805, 2014.

[18] H. R. Mendis, N. C. Audsley, and L. S. Indrusiak, "Task allocation for decoding multiple hard real-time video streams on homogeneous NoCs," in *Proceedings of the 13th International Conference on Industrial Informatics, INDIN 2015*, pp. 246–251, UK, July 2015.

[19] Y. Cao, F. Song, Q. Liu, M. Huang, H. Wang, and I. You, "A LDDoS-Aware energy-efficient multipathing scheme for mobile cloud computing systems," *IEEE Access*, vol. 5, pp. 21862–21872, 2017.

[20] P. Dziurzanski, A. K. Singh, and L. S. Indrusiak, "Feedback-based admission control for hard real-time task allocation under dynamic workload on many-core systems," in *Architecture of computing systems—ARCS 2016*, vol. 9637 of *Lecture Notes in Comput. Sci.*, pp. 157–169, Springer, [Cham], 2016.

[21] W. Wang, Y. Cao, J. Gong, and Z. Li, "CP-TPS: A Real-Time Transaction Processing Strategy Supporting Compensatory Task," in *Proceedings of the 20th IEEE International Conference on Computational Science and Engineering and 15th IEEE/IFIP International Conference on Embedded and Ubiquitous Computing, CSE and EUC 2017*, pp. 475–482, China, July 2017.

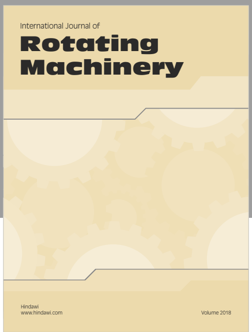
[22] H. Mendis, N. Audsley, L. Indrusiak et al., "Dynamic and static task allocation for hard real-time video stream decoding on NoCs," *Leibniz Transactions on Embedded Systems*, vol. 4, no. 2, pp. 01–25, 2017.

[23] J. Wu, C. Yuen, N.-M. Cheung, and J. Chen, "Delay-Constrained High Definition Video Transmission in Heterogeneous Wireless Networks with Multi-Homed Terminals," *IEEE Transactions on Mobile Computing*, vol. 15, no. 3, pp. 641–655, 2016.

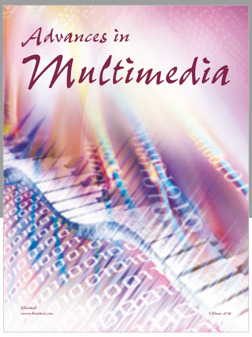
[24] J. Wu, C. Yuen, B. Cheng, M. Wang, and J. Chen, "Adaptive Flow Assignment and Packet Scheduling for Delay-Constrained Traffic over Heterogeneous Wireless Networks," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 10, pp. 8781–8787, 2016.

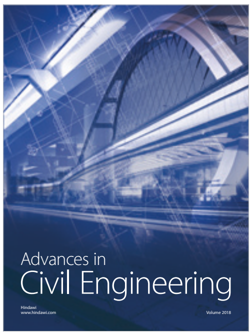
Journal of
Engineering

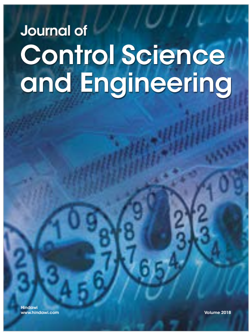
The Scientific
World Journal


International Journal of
Rotating
Machinery

Journal of
Sensors

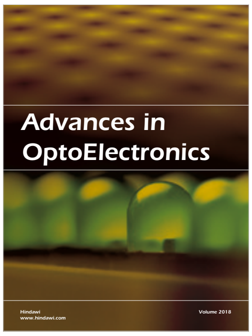

Advances in
Multimedia


Advances in
Civil Engineering


Journal of
Control Science
and Engineering

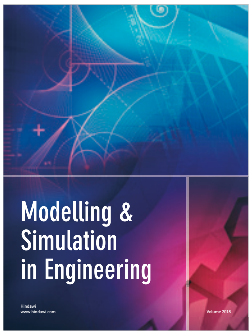
Journal of
Robotics

Journal of
Electrical and Computer
Engineering

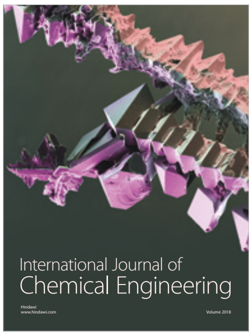

Advances in
OptoElectronics

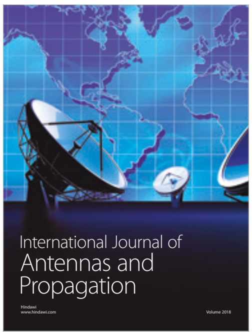
VLSI Design

International Journal of
Navigation and
Observation

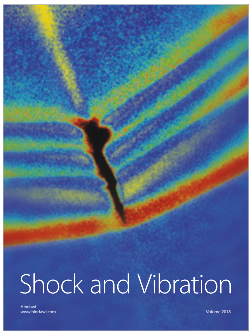

Modelling &
Simulation
in Engineering

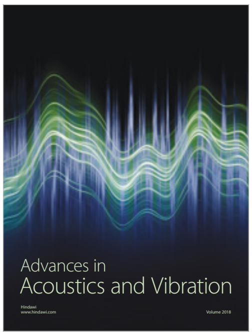
International Journal of
Aerospace
Engineering


International Journal of
Chemical Engineering


International Journal of
Antennas and
Propagation

Active and Passive
Electronic Components


Shock and Vibration


Advances in
Acoustics and Vibration