Real-Time CPU Scheduling Approach for Mobile Edge Computing System

Xiaoyi Yu, Ke Wang(✉), Wenliang Lin, and Zhongliang Deng

Beijing University of Posts and Telecommunications, Beijing, China
wangke@bupt.edu.cn

Abstract. Mobile edge computing (MEC) system has outstanding advantages of providing smart city applications with relatively low latency and immediately response. How to guarantee the QoS of the services in MEC system is consequently becoming a hot issue. This work focuses on solving the problem by real-time CPU scheduling. The proposed scheduling algorithm considers different services arrival profiles, computation time consumption and deadline requirements simultaneously. Specifically, the combination and optimization of support vector machine (SVM) and earliest deadline first (EDF) algorithm is designed, which could automatically classify services type and efficiently allocate the computation time in real-time manner. By deploying the traffic trace from the real world, the proposed scheduling algorithm could reduce 45% latency and improve the reliability of transmission, comparing with popular fixed-priority CPU scheduling algorithm.

Keywords: SVM · EDF · Low latency
High reliability of transmission

1 Introduction

Mobile Edge Computing (MEC) System, a concept proposed by ETSI in 2014, provides IT and cloud computing capabilities within the radio access networks in close proximity to mobile subscribers [1]. It is widely considered to be the key technology to realize future smart city applications, such as wireless smart grid, Internet of things, etc.

MEC reduce the latency by offloading computation-intensive tasks to the edge cloud. However, the limited computational resource in the edge clouds may result in the Quality of Service (QoS) degradation [2]. Multi-class network traffic classification helps identify the application utilizing network resources, and facilitate the instrumentation of QoS for different applications. Early traffic classification systems rely on transport layer port number to classify flows. However, with the wide use of dynamic ports, the less effectiveness makes the technique based on port number unreliable. Signature matching technique was proposed by Moore [3]. It derives signature patterns from various network traffic flows and classifies the traffic flow through these matching signature patterns. Although

P. H. J. Chong et al. (Eds.): SmartGIFT 2018, LNICST 245, pp. 32–42, 2018.
https://doi.org/10.1007/978-3-319-94965-9_4

it's classification accuracy is high, the continuous updating of signature patterns and its inability of handling encrypted packets limit the application [4]. Machine learning methods classify traffic flows according to the flow's statistical characteristics (e.g. packet size, flow duration, etc.). In [5], the authors use 12 features for two data sets, the UNB ISCX network traffic data set and their internal data set, to classify by k-NN classification algorithm. Tsinghua university in [6] classify 7 classes of internet applications with 9 feature parameters, and all of them can be obtained from the packet header. These methods provide a guideline to classify the network traffic, but lots of features also increases processing time, leading to serious latency.

In a real-time system, system's performance and throughput are highly affected by CPU scheduling. With the development of smart phones and wearable devices, the problem of finding robust or flexible solutions for scheduling problems is very importance for applications. Out of some important scheduling, Round Robin algorithm is much efficient, which assume that all servers have the same processing performance. An efficient dynamic Round Robin algorithm for CPU scheduling in [7]. However,this scheduling algorithm depends on choosen time quantum and the relationship between time quantum size and process running time. [8] provide a algorithm, which is based on the existing EDF dynamic scheduling algorithm. The algorithm improves the real-time response of EDF to a certain extent by using the comparison of priority and the time slice borrowing strategy, which is disadvantageous to the low utilization rate of idle time slice of EDF algorithm. In fact, the CPU scheduling algorithm for real-time tasks with deadline has been extensively studied in real-time systems. Dynamic-priority-based EDF algorithm is known to be theoretically optimal for scheduling sporadic real-time tasks [9].

To the best of our knowledge, there is no real-time scheduling algorithm for the multi-class network traffic with SVM classification algorithm in MEC system has been proposed. And in existing studies, most of them are just using SVM for basic classification. There are few optimizations for its parameters. In this paper, we will put forward a real-time CPU Scheduling approach for Mobile Edge Computing System, which combine SVM classification algorithm with parameters optimization and EDF Scheduling algorithm for Mobile Edge Computing. 4 classes of network traffic with 2 feature parameters is used to classify. The simulation results show that combining the SVM algorithm and EDF algorithm can reduce computing latency about 45% and improve the reliability of transmission throughput compering with FP scheduling.

2 System Model

2.1 C-RAN Framework Model

C - RAN network evolve from the traditional distributed base station, as a new type of broadband wireless Internet access technology. We choose C - RAN network as an example to introduce our experiment.

Consider a general C-RAN system which consists of three main components: remote radio heads (RRHs), baseband processing units (BBUs), optical transport network. As shown in Fig. 1, there are N RRHs serving N cells in the transmission system and there are M BBUs in the BBU pool.

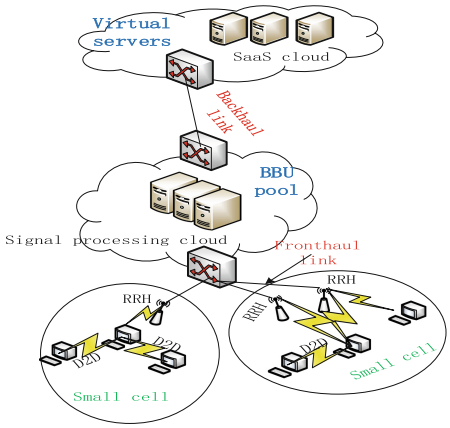

Fig. 1. c-ran framework.

In the process of the BBU pool, PRB increase with the MCS. Then we will build an accurate model to describe the contribution of each underlying BBU functions to the total processing time and how they scale with the increase of PRB and MCS. In this experiment, we consider the three main BBU function: IFFT/FFT, modulation and coding. Generally, there are two important elements in the BBU pool in the course of processing: basic processing and dynamic processing load. The basic processing includes IFFT and FFT for each PRB and the platform-specific processing relative to the reference GPP platform. The dynamic processing load includes user processing, namely coding and modulation, which is the distribution of the PRBs and the linear function of the MCS. On this basis, a model for calculating the total BBU processing time of different PRB, MCS and platforms is proposed, and the following formula is presented as

$$T_{subframe}(z,r,w)[us] = \underbrace{c[z]+p[w]}_{baseprocessing} + \underbrace{u_r[z]}_{RMSE} + \underbrace{u_s(z,r)}_{dynamicprocessing} . \quad (1)$$

where the triple (z; r; w) represents PRB, MCS, and platform. The c[z] and p[w] are the base offsets for the cell and platform processing, $u_r[z]$ is the reminder of user processing, and $u_s(z,r)$is the specific user processing that depends on the allocated PRB and MCS. The $u_s(z,r)$ is linearly fitted to $a(z)r+d(z)$, where a, d are the coefficients, and r is the MCS. Table 1 provide the processing model parameters of the Eq. (1).

However, in our experiment, we chose the LXC as our experiment platform. Considering the practicability of the C - RAN network, the number of PRB that each RRH set configuration is determined by system bandwidth and assigned to each user equipment PRB is derived by the channel status. We assume that these two amount of PRBs are static. Under this promise, we set PRB to 25. Note that MCS 9, 16, and 27 corresponds to QPSK, 16QAM, and 64QAM with the highest coding rate [10]. In our experiment, we randomly picked one of the three values to be the MCS value.

Table 1. Processing model parameters in u_s

z	c	p		$u_s(z,r)$		u_r	
		GPP	LXC	a	d	GPP	LXC
25	23.81	0	5.2	4.9	24.4	41.6	57.6
50	41.98	0	5.7	6.3	70	79.2	80

2.2 Traffic Model

The Professional term and symbol in this work are following the traditional definition in real-time systems. we adopt the most commmonly used traffic model for introducing the real-time analysis into C-RAN, i.e. periodic task with constrained deadline. In each RRH, we assume that there are four types of traffic:video, Browse the web, qq, e-mail. In this section, we use an array of P elements (x_p,y_p) and a mapping f(x) → y to describe the packet that we capture, we define X = $\{x_1, x_2, \ldots, x_p\}$ as our traffic flow set, flow x_p properties for classification, $x_p = \{x_{pq} | q = 1\ or\ 2\}$, q is the number of attributes to class, x_{pq} said the p_{th} packet of the q_{th} properties. Y = $\{y_1, y_2, y_3, y_4\}$to classify categories, respectively, qq, browse the web, video, e-mail. We selected P1 packets from P data packets as training set data for classification function of training.

In our system, we choose four type traffic flow to be observed. We selected two attributes of the data packet, time delta from previous captured frame is defined as t_p, another kind is the length of the packet l_p, among them (p = 1, 2,..., P). P is the total number of packets we capture.t_p and l_p is the x_{pq} which we mentioned above.

First of all, we only run a type of task in the computer, and then capture the network port information. The length of the packet and time delta from previous captured frame of four kinds of network traffic task is shown in Fig. 2. From the Fig. 2, we can clearly see the two attributes of the each task have the obvious difference. The average packet length of video is smaller than the rest. The time delta from previous captured frame of qq is the smallest. The e-mail has the longest packet and browseing the web has its own characteristics too. So, we choose these two properties as attributes for our classification.

For a task x_p, we can use AT_p to express the arrival time (a task start preparing processing time). The deadline is defined as DT_p (the task must be

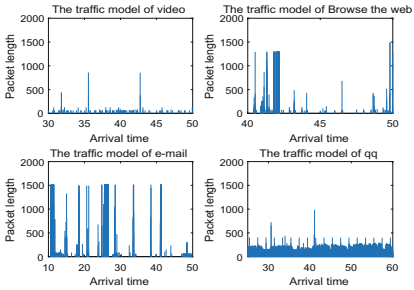

Fig. 2. The traffic model of the four type of task.

completed before the time). For each task x_p, we set a remaining run time RT_p, $RT_p = DT_p$ - t, where t is the current time. For each task x_p, minimum of the RT_p has a higher priority obviously. Obviously, the task x_p which has a higher priority can be processed firstly. x_p can be characterized by three positive integers-worst case execution time $WCET_p$, deadline DT_p, and cycle CT_p, where $WCET_p < DT_p < CT_p$. We set the average of the time delta from previous captured frame as the cycles of four kinds of task. The cycle CT_p of each task is shown in Table 2.

Table 2. Processing model parameters in u_s

Type	qq	Video	e-mail	Browse the web
Cycle/us	0.009203	0.002664	0.003743	0.025081

3 SVM Based Traffic Classification

3.1 SVM Algorithm

We choose SVM as our classification algorithm is because the algorithm can minimize the empirical classification error and maximize set edge classification space. These features reduce the excessive learning the structure of the risk in the limited samples.

Each of our data packets can be expressed as a point on the axis, which can be separated by a line or a plane. We assume that this line or plane is

$$f(x) = w \cdot x + b \tag{2}$$

However, dividing the tasks with only two attributes into four categories can not be done in two-dimensional space, so the kernel function is used in the SVM algorithm. The common kernel functions are linear, polynomial, RBF, etc. We

used the RBF kernel function in this experiment. In a multidimensional space, $\mathrm{x} \rightarrow \varphi(\mathrm{x})$, RBF kernel function can be represented as:

$$\mathrm{K}\left(x_i, x_j\right) =< \varphi\left(x_i\right), \varphi\left(x_j\right) >= \exp\left(-g\left\|x_i - x_j\right\|^2\right). \tag{3}$$

In order to be able to more accurately classified. We can reduce the problem to:

$$\begin{cases} \min\limits_{w,b} \ \frac{1}{2}\,\omega^2 + C\sum\limits_{p=1}^{p1}\vartheta_p \\ s.t \\ y_p\left(\omega \cdot \varphi\left(x_p\right) + b\right) \geq 1 - \vartheta_p \\ \vartheta_p \geq 0 \end{cases} \tag{4}$$

The objective function is to maximize the distance between the data points, where ϑ_p is slack variables. The corresponding is that data points x_p allow deviation from the amount of hyperplane. C is penalty coefficient that can limit ϑ_p to infinity. The problem can be solved by Lagrange multiplier. Then Eq. (4) can be converted into Eq. (5) on the dual problem.

$$\begin{cases} max \sum\limits_{p=1}^{P1}\alpha_p - \frac{1}{2}\sum\limits_{I}^{P1}\sum\limits_{J}^{P1}\alpha_i\alpha_j y_i y_j K\left(x_i, x_j\right) \\ s.t \quad \sum\limits_{p=0}^{P1}\alpha_p y_p = 0 \quad 0 \leq \alpha \leq C \end{cases} \tag{5}$$

where α_p is the parameter of Lagrange multiplier. By Eq. (5), we can get w and b. The decision function of the final result can be expressed as:

$$\mathrm{f}\left(\mathrm{x}\right) = \omega \cdot \mathrm{x} + \mathrm{b} = \left[\sum_{p=1}^{P1}\alpha_p y_p K\right] + \mathrm{b} \tag{6}$$

In the way, The value of f(x) is represented as one of the type of qq, browse the web, video and e-mail.

3.2 SVM Parameter Optimization

To use SVM, we need to set the parameters. From the above know, we select the RBF kernel function. For RBF kernel functions, we need two functions in general: C and g. For a given problem, we don't know the optimal number of C and g in advance, so we have to search parameter to find the optimal (C, g).

In this experiment, we use the method of grid searching to find the optimal parameters. Because the parallelism of the grid search is very high, and each parameter is independent of each other. The variation range of the penalty coefficient C is $\left[2^{c\,\min}, 2^{c\,\max}\right]$, where we can look for the best C. The default value is $c\min = -8, c\max = 8$. Similarly, the variation range of g is $\left[2^{g\,\min}, 2^{g\,\max}\right]$ and the default value is $g\min = -8, g\max = 8$. C and g is the horizontal and vertical axes of the grid, cstep and gstep are the step sizes of C and g, which

are optimized by the grid parameters. The default values of step size are 1. In this case, The value of C is $\left[2^{c\min}, 2^{c\min}+1, ...2^{c\max}\right]$ and the value of g is $\left[2^{g\min}, 2^{g\min}+1, ...2^{g\max}\right]$.

The grid search is to try every possible parameters and use each (C, g) to classify. And then find the best (C, g), which possess the highest precision of cross validation.

Parameter C controls the largest hyperplane and minimizes the data point deviation. We set C to 32768 and g to 8 by cross-validation in our experiment and the accuracy of cross-validation was 79.9729%.

Algorithm 1. Parameter Optimization Algorithm

```
c = g = 2^-8, m = 0
while C < 2^8 do
    C = 2^-8 + m, m = m + 1, n = 0
    while g < 2^8 do
        g = 2^-8 + n, n = n + 1
        Use the current g and C for classification. Calculate and record classification accuracy, C and g.
    end while
end while
To sort the classification accuracy.
return C and g with the highest classification accuracy.
```

4 Traffic Schedule EDF

In this section we will introduce a preemptive EDF scheduling algorithm.

In numerous real-time scheduling algorithm, the scheduling algorithm that based on priority is one of the most important type of scheduling algorithm in real-time scheduling method. According to the different priority assignment strategy, the scheduling algorithm can be divided into static priority scheduling and dynamic priority scheduling. In general, the dynamic priority scheduling algorithm is better than static priority scheduling algorithm. EDF algorithm is a typical representative of the dynamic priority scheduling algorithm. So we choose EDF algorithm as our scheduling algorithm.

Preemptive EDF scheduling algorithm always performs the first real-time task of the deadline. It is a dynamic priority scheduling algorithm, which is based on the following assumptions:

(1) There is no unpreemptible part of any task, and the cost of preemption can be ignored;

(2) Only the processor requests make sense, memory, I/O, and other resources requests can be ignored;

(3) All tasks are irrelevant; There is no constraint of order;

Based on the assumption of above (1)–(3), the necessary and sufficient condition of the EDF scheduling algorithm for a given periodic task set scheduling is:

$$U = \sum_{p=1}^{P} \frac{T_{subframe}}{CT_p} \leq 1 \tag{7}$$

Thus, the biggest advantage of the preemptive EDF scheduling algorithm is that, for any given set of tasks, as long as the processor utilization is not more than one hundred percent, it can guarantee its scheduling.

As described above, the task x_p is defined by the tuple $\{WCET_p, DT_p, CT_p\}$, Therefore, we need to define how to calculate these parameters in the following. CT_p is shown in Table 2.

$$\begin{cases} WCET_p = T_{subframe}(z, r, w)[us] \\ DT_p = AT_p + T_{subframe}(z, r, w)[us] \end{cases} \tag{8}$$

Algorithm 2. EDF scheduling Algorithm

Input:
CT_p, t(t is the current time), $T_{subframe}$

When a data packet arrives, read the arrival time AT_p in the packet
Calculate $DT_p = AT_p + T_{subframe}(z, r, w)[us]$
Calculate $U = \sum_{p=1}^{P} \frac{T_{subframe}}{CT_p} \leq 1$
while $U \leq 1$ **do**
    if $t = AT_p$ **then**
        put the data packet into the pending sequence
when no new data packet arrives
According to DT_p sort from small to large.
Deal with the packet with the smallest deadline, the rest packet wait next scheduling
return The running task

5 Simulation

In this part, we study the influence of SVM algorithm and EDF algorithm on the real-time CPU Scheduling approach for Mobile Edge Computing System by simulation.

In the process of classification, We capture 10G data packets by using wireshark through server which network port rate is 5M. The packets contain the four types of task that we will classify. Using cross validation for parameters optimization can improve the classification accuracy of the data. The classification accuracy of the four types of task that we choose is shown in the following Table 3. These tasks is classified by SVM algorithm which use default parameters and parameters optimization respectively.

Table 3. The influence of parametric optimization on classification accuracy

Type	qq	Video	e-mail	Browse the web
The number of packet	196710	849613	22771	63778
Classification accuracy of parameter optimization	0.88726	0.85094	0.84923	0.831525
Classification accuracy of default parameters	0.803594	0.85	0.83536	0.827731

From the Table 3, we can see clearly that using parameter optimization can improve the accuracy of classification.

For scheduling algorithm, in contrast, we select a fixed priority scheduling algorithm, which the scheduling priority of our task is set by the people. In this experiment, we set video as the highest priority, the second is Browse the web, E-mail is the third priority, priority of qq is the last.

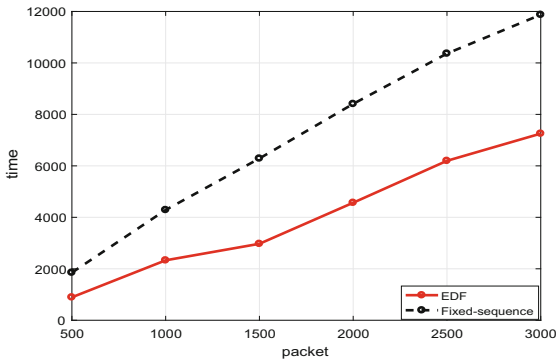

Fig. 3. The influence of different scheduling algorithms on processing time.

In Fig. 3, when we adopt the parameter optimization of SVM classification algorithm and transfer the same number of the packet, we can see that EDF algorithm is better than the fixed priority scheduling algorithm greatly to reduce the time. At the same time, the effectiveness of the EDF algorithm can cut down the processing time and delay, which meet the requirements of transmission.

In Fig. 4, comparing the two kinds of scheduling algorithm of packet loss rate can see clearly that EDF algorithm of packet loss rate is much lower than the fixed priority scheduling algorithm of packet loss rate, which ensure the reliability of transmission.

The simulation results show that the combination of SVM algorithm and EDF algorithm can effectively improve the efficiency of transmission system, satisfy the high reliability and low delay requirement of the requirements of 5G and IOT.

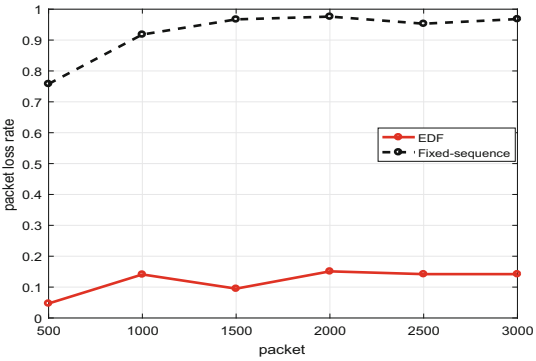

Fig. 4. The influence of different scheduling algorithms on packet loss rate.

6 Conclusions

In this paper, we proposed a real-time CPU Scheduling Approach for Mobile Edge Computing System. It attempts to reduce the latency of the transmission system and the packet loss rate by combining SVM and EDF, and provides a new angle to the scheduling algorithm. The simulation results have illustrated the efficiency of the algorithm.

References

1. Mao, Y., et al.: A survey on mobile edge computing: the communication perspective. IEEE Commun. Surv. Tutor. **19**, 2322–2358 (2017)
2. Zhao, T., et al.: Tasks scheduling and resource allocation in heterogeneous cloud for delay-bounded mobile edge computing. In: 2017 IEEE International Conference on Communications (ICC). IEEE (2017)
3. Jing, N., et al.: An efficient SVM-based method for multi-class network traffic classification. In: 2011 IEEE 30th International Performance Computing and Communications Conference (IPCCC). IEEE (2011)
4. Hao, S., et al.: Improved SVM method for internet traffic classification based on feature weight learning. In: 2015 International Conference on Control, Automation and Information Sciences (ICCAIS). IEEE (2015)
5. Yamansavascilar, B., et al.: Application identification via network traffic classification. In: 2017 International Conference on Computing, Networking and Communications (ICNC). IEEE (2017)
6. Li, Z., Yuan, R., Guan, X.: Accurate classification of the internet traffic based on the SVM method. In: IEEE International Conference on Communications 2007, ICC 2007. IEEE (2007)
7. Farooq, M.U., Shakoor, A., Siddique, A.B.: An Efficient dynamic round robin algorithm for CPU scheduling. In: International Conference on Communication, Computing and Digital Systems (C-CODE). IEEE (2017)
8. Yue, M., Yue-Qi, Z., Zhen-Yu, Y.: Research on real-time scheduling method of RTAI-linux based on edf algorithm. In: 2017 10th International Conference on Intelligent Computation Technology and Automation (ICICTA). IEEE (2017)

9. Pathan, R.M.: Design of an efficient ready queue for earliest-deadline-first (EDF) scheduler. In: Proceedings of the 2016 Conference on Design, Automation and Test in Europe. EDA Consortium (2016)
10. Nikaein, N.: Processing radio access network functions in the cloud: critical issues and modeling. In: Proceedings of the 6th International Workshop on Mobile Cloud Computing and Services. ACM (2015)