Wireless Networks
https://doi.org/10.1007/s11276-018-1892-z

Computing aware scheduling in mobile edge computing system

Ke Wang[1] · XiaoYi Yu[1] · WenLiang Lin[1] · ZhongLiang Deng[1] · Xin Liu[1]

Abstract

Mobile edge computing (MEC) is an emerging technology recognized as an effective solution to guarantee the Quality of Service for computation-intensive and latency-critical traffics. In MEC system, the mobile computing, network control and storage functions are deployed by the servers at the network edges (e.g., base station and access points). One of the key issue in designing the MEC system is how to allocate finite computational resources to multi-users. In contrast with previous works, in this paper we solve this issue by combining the real-time traffic classification and CPU scheduling. Specifically, a support vector machine based multi-class classifier is adopted, the parameter tunning and cross-validation are designed in the first place. Since the traffic of same class has similar delay budget and characteristics (e.g. inter-arrival time, packet length), the CPU scheduler could efficiently scheduling the traffic based on the classification results. In the second place, with the consideration of both traffic delay budget and signal baseband processing cost, a preemptive earliest deadline first (EDF) algorithm is deployed for the CPU scheduling. Furthermore, an admission control algorithm that could get rid off the domino effect of the EDF is also given. The simulation results show that, by our proposed scheduling algorithm, the classification accuracy for specific traffic class could be over 82 percent, meanwhile the throughput is much higher than the existing scheduling algorithms.

Keywords MEC · SVM · EDF · Scheduling · Admission control

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. Driven by this concept, in recent years the trend of increasingly moving the cloud computing towards the network edge is observed. It is estimated that tens of billions of Edge devices will be deployed in the near future, and their processor speeds are growing exponentially, following Moore's Law [1]. The increasingly computation load caused by devices makes the MEC designing should following both disciplines of computing and wireless communications.

One typical MEC system in the cellular communication systems, such as 4G, is co-located with the cloud radio access network (C-RAN) [1]. C-RAN divides the traditional base station into three parts, i.e., remote radio heads (RRHs), baseband unit (BBU) pool, and the fronthaul link. The RRHs only need to compress and forward the received signals to BBU pool or transmit wireless signals to devices. Whilst most of the intensive network computation tasks, such as baseband signal processing, precoding matrix calculation, channel state information estimation are moved to the BBU pool. In this context, the MEC functions are integrated into the BBU pool, responsible for the computation resource-hungry applications and performance improvements such as video coding, traffic classification, scheduling. However, comparing with traditional centralized cloud in the core network, the BBU pool has much less computation resources. Therefore, one key issue in designing a multi-user MEC system is how to allocate the finite radio-and-computational resources to multiple mobiles to guarantee the QoS of diverse applications.

In this work, we consider a BBU pool with multiple mobile devices, each device has one traffic flow. We

✉ Ke Wang
wangke@bupt.edu.cn

[1] Key Laboratory of Universal Wireless Communications Ministry of Education, Beijing University of Posts and Telecommunications, Beijing, People's Republic of China

Published online: 01 January 2019

mainly focus on the downlink traffic transmission, where the traffic from the core network firstly enter the BBU pool, then by deploying the scheduling algorithm, the CPU of BBU servers transform the packets into the baseband signals and transmit them to the RRHs. The BBU server could perform the traffic monitoring, CPU monitoring and channel state information (CSI) analyzing. Under this premise, we solve the issue by proposing a computing aware scheduling algorithm which includes two parts: real-time traffic classification and preemptive EDF scheduling.

To demonstrate proposed scheduling algorithm, we need to clarify the scheduling procedures defined in LTE [2]. When a traffic flow arrives at the BBU, a dedicated bearer between the BBU and device is set up. Depending the QoS requirements, the bearers can be further classified as Guaranteed-bit-rate (GBR) or non-guaranteed-bit-rate (non-GBR). In this context, the general definition of QoS requirements is translated in variables that characterize performance experienced by users. A set of QoS parameters is therefore associated to each bearer depending on the application data it carries, thus enabling differentiation among flows. The LTE specify several QoS classes which is identified through QoS class identifiers (QCIs) [3], i.e., scalar value use as a reference for driving specific packet forward behaviors. Each QoS class is characterized by its resource type (GBR or non-GBR), a priority level, the delay budget and acceptable packet loss rate. The radio resource management (RRM) module translates QoS parameters into scheduling parameters, admission policies, queue management thresholds, link layer protocol configurations, and so on [4]. The scheduler of BBU then performs the traffic scheduling and resource allocation. The resource allocation including select the proper modulation and coding scheme (MCS) and physical resource block (PRB) according to the devices' CSI.

In general, the standardized LTE scheduler makes its decision with the consideration of (1) QoS requirement of different traffic class; (2) the PRBs' CSI of associated devices. However, there are two main drawbacks in practice: Firstly, there is no label in the traffic packets to indicate which traffic class the packets belong to. So the widely used scheduler in BBU, such as, round robin, maxCQI [5], M-LWDF and EXP/PF [6] are not taking traffic class into account, hence cannot guarantee the divers QoS requirements. Secondly, the LTE scheduler does not consider the computation load in resource allocation. Such scheduler is efficient when the traffic load is low, however, with increasingly amount of devices, this assumption will no longer exist in most of scenarios. As shown in [7], the latency caused by MEC and baseband signal processing of C-RAN both affect the traffic's QoS. This work aims to tackle the above drawbacks by introduced machine learning and real-time CPU scheduling algorithms into the scheduler design. The main contribution can be summarized as follows:

- We provide a SVM based traffic classifier, the parameter tunning and training steps are discussed in detailed. Furthermore, we also give out the results of feature selecting experiment, which considers seven feature combinations with 8 types of features. The feature selecting considers both accuracy and algorithm complexity. The given traffic classifier could real-time classify 4 most common class of traffic (web surfing, video stream, OICQ and E-mail) with accuracy above 83 percent.
- We provide a preemptive EDF CPU scheduler, which could not only consider the diverse QoS requirement of different traffic class, but also consider the baseband signal processing load. Moreover, a computing aware admission control is designed to avoid the domino effect of EDF scheduling algorithm. The proposed EDF with admission control could obtain lower packet drop rate and comparable computing complexity with non-admission control EDF algorithms.
- To the best of our knowledge, no real-time scheduling algorithm for the multi-class network traffic with SVM classification algorithm in MEC system has been proposed. This work gives a first shot in this field, and the proposed framework could be easily extend to other traffic classification and CPU scheduling algorithms.

The remainder of this paper is organized as follows. Section 2 introduces several related works. In Sect. 3 we gives out the system model, traffic model and baseband processing model, respectively. In Sect. 4, we discuss the details on real-time traffic classifications. In Sect. 5, the scheduler with admission control is given. In Sect. 6, we provide numerical results through simulations. Finally, this paper is concluded with Sect. 7.

2 Related work

The joint radio-and-computational resources management plays a important role in realizing the energy-efficient and low-latency MEC. In this Section, we will first review the existing resource allocation and scheduling schemes in MEC, then introduce the current research progress in traffic classification, at last summarize the results of real-time CPU scheduling.

Recent proposed scheduling algorithms for wireless communication systems, such as LTE, mainly consider the queue status, channel condition, antenna technologies, etc., in a cross layer manner [4]. Very few of these algorithms take the computation resource into account. However, how to take full advantage of the computation resources in MEC is a major challenge, since the BBU pool needs to support hundreds of times more user equipments (UEs) comparing with one single base station. In addition, with explosive increase of multi-media traffic, how to meet the hard deadline constraint is another challenge to scheduling algorithm design in C-RAN [8].

Unsurprisingly, considering the computation requirement in C-RAN based MEC problems is beginning to receive attention. In [9, 10], the authors show a computation outage probability in C-RAN, and also prove that the choice of modulation and coding schemes (MCS) highly affect the computational requirements. In [11], a system power minimization scheme based on VM assignment and cooperative transmission is proposed. In [12, 13], two different beamforming algorithms with considering of computation efforts (in Giga Operations Per Second) are designed respectively. All of these excellent works use the computation capacity to formulate the optimization problem. Based on their results, a guideline to design the system is well presented, but how to use the computation resources in real-time is still unknown, which needs to be solved by scheduling algorithms.

The MEC scheduling algorithms always assume the arrival of different users are in general asynchronous so that it is desirable for the edge server with finite computation resource to buffer and compute the tasks sequentially, which incurs the queueing delay. In [14], to cope with the bursty task arrivals, the server scheduling was integrated with uplink downlink transmission scheduling to minimize the average latency using queuing theory. Second, even for synchronized task arrivals, the latency requirements can differ significantly over users running different types of applications ranging from latency-sensitive to latency-tolerant applications. This fact calls for the server scheduling assigning users different levels of priorities based on their latency requirements. In [15], after the pre-resource allocation, the MEC server checked the deadline of different tasks during the server computing process and adaptively adjusted the task execution order to satisfy the heterogeneous latency requirements. Last, some computation tasks each consists of several dependent sub-tasks such that the scheduling of these modules must satisfy the task-dependency requirements. The task model with a sequential sub-task arrangement was considered in [16] that jointly optimized the program partitioning for multiple users and the server computation scheduling to minimize the average completion time.

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 [17]. It derives signature patterns from various network traffic flows and classifies the traffic flow through these matching signature patterns. Although its classification accuracy is high, the continuous updating of signature patterns and its inability of handling encrypted packets limit the application [18]. Machine learning methods classify traffic flows according to the flow's statistical characteristics(e.g. packet size,flow duration,etc.). In [19],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. In [20] the authors 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.

The core of real-time computational scheduling is the design of the scheduler in operation system, which could precisely control the CPU's behavior. Both of the test-bed based studies in [8, 21] provide a CPU processing model in C-RAN, whereby [8] has also provided a CPU scheduling algorithm for RRHs. However, this scheduling algorithm does not count for the variation of computation requirement among the UEs. By using real-world cellular traffic, [8] has found each CPU could serve at least 4 cells with heavy traffic load. In fact, the CPU scheduling algorithm for real-time tasks with hard deadline has been extensively studied in real-time systems [22, 23]. Specifically, by the extensive empirical comparison, [23] has proved the following facts for real-time tasks with hard deadline: (1) the partitioned scheduling algorithm (which means each task is statistically assigned to a single processor with no migration allowed) is consistently better than global approaches (which are contrast to partitioned scheduling); (2) the deadline based scheduling algorithm performs better than other scheduling algorithms.

3 System model and back ground

3.1 Generalized MEC model

The generalized mobile edge computing model is given in Fig. 1. For the purpose of easier description, we extract the functions of the mobile edge computing system into three network function entities (i.e. classifier, scheduler, baseband processing server), respectively. In practice, these functions can be virtualized and sequently deployed in an independent edge computing server. As shown in the figure, the mixed traffic flows are entering the MEC by firstly going through the traffic classifier. By which, the packets are classified into predefined traffic types. Hence the

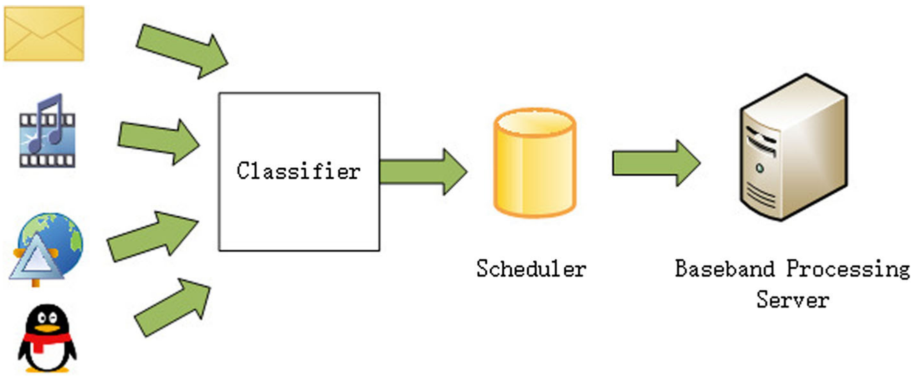


Fig. 1 Generalized MEC model

quality class identifier (QCI) of each packet is decided. Based on the results of the classifier, the scheduler checks the QCI and find out the delay budget of each packet. The admission control is then deployed by the scheduler with the consideration of QoS requirement and current computing resource utilization status. After that, the scheduler deploys the scheduling algorithm to select the packet that will be served next, and delivers it to the next entity. The baseband processing server receives the packets from the scheduler. It is primarily responsible for the baseband signal processing functions, such as: modulation and coding, IFFT, etc,. Other parts of the MEC which include upconvertion and wireless transmission is not in the scope of this article.

3.2 Traffic model

The main focus of the studies that worked on the network traffic classification is to classify the traffic including web surfing, Instant Messaging (such as QQ), E-mail and video. In order to make the research representative, we have selected four classes representative traffic classification which are HTTP/HTTPS, OICQ, SMTP and RTSP, corresponding the above traffic separately.

The four classes network traffic can be represented as a set of stream, and the stream is consists of packets. It is worth noting that all the packets are defined by numerous features (e.g., source port, destination port). The ith packet can be identified by a vector $\boldsymbol{x}_i \in R^n, i = 1, 2, \ldots, l$, and a label $y_i \in R^l$ that indicates which class the packet belongs to.

The traces of the four classes packet are shown in Fig. 2. If a packet has the following same features: protocol, source address, destination address, source port and destination port, it will be labelled to the same stream. Figure 2(a) shows the trace of the RTSP. It can be clearly seen from the figure that the stream of the RTSP has more packets than other classes traces. The RTSP can be divided into two state, which are service beginning state and service steady state. During the service beginning state, the packets are detected. When the state transforms into service steady state, packets start to be transmitted. Figure 2(b) shows the trace of the HTTP/HTTPS. Since the HTTP/HTTPS connection is held for the duration of a random number of request/response transaction, the stream of HTTP/HTTPS is intermittent and the inter-arrival time is chaotic. Figure 2(c) shows the trace of the SMTP. In general, when the user uses the E-mail, the headers of the available messages are downloaded to the computer from the server. The user will then scan through the headers and download the messages that the user require. When the user finishes with the current message, the user will deal with the next message. Hence, the stream of the SMTP represents the user's download time, and the interval between two consecutive streams represents the user's reading time. Figure 2(d) shows the trace of the OICQ. The OICQ is made up of intermittent bursts, and each burst is consisted of a stream. The reason for this situation can be explained by following fact: an interval of continuous talking is a talkspurt, which results in a burst of consecutive packets.

Figure 3 illustrates an example that the ith packet is scheduled in CPU at t. The length of the packet represent the processing time of the packet. When the packet arrives, it waits to be scheduled. Up arrows and down arrows denote the arrival time and deadline of the packet. The arrival time of ith packet is written as AT_i. Then the inter-arrival time between the ith packet and $(i+1)$th packet can be denoted by IT_i. Moreover, DT_i stands for the deadline. The processing time of the ith packet can be denoted as T_i, and then the maximum processing time is written as PT_i. The remaining processing time can be defined as ST_i. In addition, the remaining maximum processing time RT_i can be calculated by:

$$RT_i = DT_i - t \tag{1}$$

According to LTE standard [24], the ith packet delay budget of network in this work is denoted as D_i. The delay budget of network is consisted of two parts: transmission time TT_i and maximum processing time PT_i. The transmission time of the packet is the ratio of packet length to

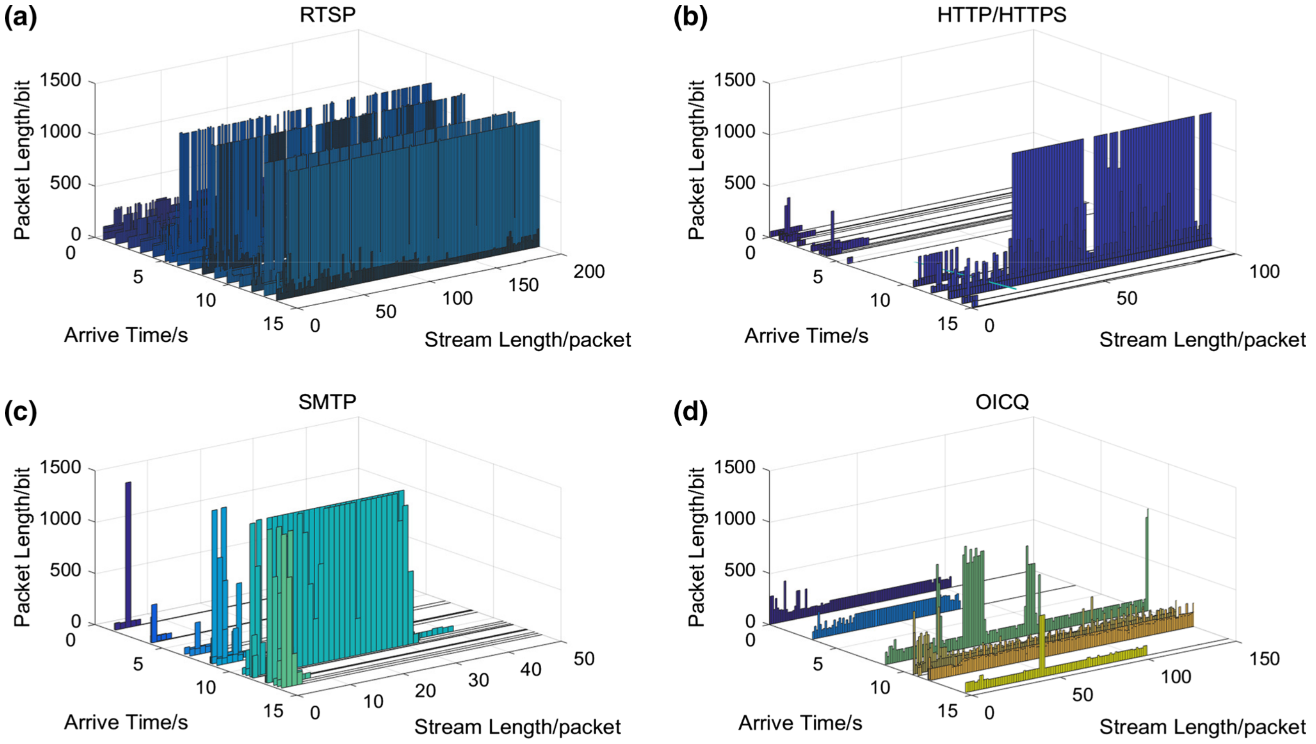
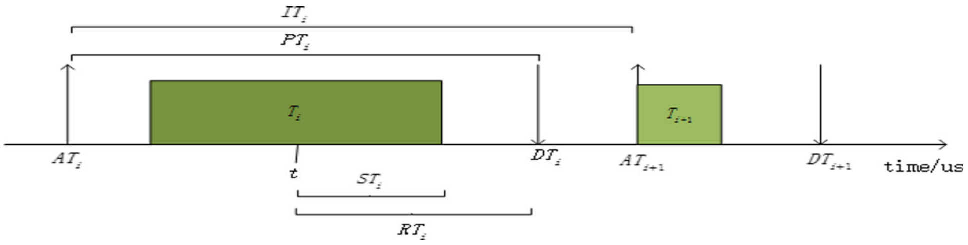


Fig. 2 Traffic trace

Fig. 3 An example of the packet time series in CPU



bit rate (which will be discussed in next subsection). Hence, we can deduce the DT_i by:

$$DT_i = AT_i + D_i - TT_i. \tag{2}$$

3.3 Baseband processing model

In this work, we only concentrate on the network traffic classification and scheduling. We assume the processing time is only related to the factors, which are considered including (de)coding, (de)modulation and FFT/IFFT. The processing of the packet is fully implemented in the BBU pool. Under this promise, the value of MCS is chosen randomly.

For each factor, the processing time is measured on the Intel E5-2600V4\DDR4 64G\ Broadcom NetXtreme Gigabit Ethernet for different PRB z_i and MCS r_i. The FFT/IFFT increase only with the PRB, while (de)coding and (de)modulation are increase as a linear function of allocated MCS and PRB. In addition, the (de)coding represents the most time consuming factors in processing.

According to [8], the processing includes two components: base processing and dynamic processing. Based on the above introduction, the base processing is consisted of FFT/IFFT. Considering the practicability of the processing model, the base processing only depends on the channel bandwidth and is imposed a constant load on the system. On the other side, the dynamic processing load includes (de)coding and (de)modulation.

In this work, we assume these PRBs are 25. Note that different values of MCS correspond to different modulation modes which are shown in Table 2. We can deduce a model to calculate the actual processing time for different MCS by:

$$T_i(z_i, r_i)[us] = \underbrace{c[z_i]}_{\text{base processing}} + \underbrace{u_s(z_i, r_i)}_{\text{dynamic processing}}. \tag{3}$$

where the triple (z_i, r_i) represents PRB, MCS. The $c[z_i]$ is the base processing and $u_s(z_i, r_i)$is the dynamic processing that depends on the allocated PRB and MCS. The $u_s(z_i, r_i)$ is linearly fitted to $a(z_i)r_i + b(z_i)$, where a, b are the coefficients and r_i is the MCS. Table 1 provides the processing model parameters of the Eq. (3).

Considering the LTE Turbo-coder, the internal interleaver is decided by a limited number of code-block sizes and the set of transport block size is defined in LTE to ensure the transport block size (i.e. the bits carried by PRBs) of arbitrary size which can be segmented into code blocks that match the set of available code-block sizes [25]. Based on the MCS index and the number of allocated PRB, the transport block size can be obtained by looking up the Tables 7.1.7.1-1 and 7.1.7.2.1-1 provided in specification [26]. According to the protocol [26], the transmission time of each block equal to 1 ms. Therefore, the bit rate and the corresponding symbol rate can be obtained. Different maximum processing times PT_i corresponding to different MCS are shown in Table 2.

Table 2 Different values of MCS correspond to rate and maximum processing time

MCS	9	16	27
Modulation Modes	QPSK	16QAM	64QAM
Block Size	4008	7736	15840
Bit Rate /(Mbit/s)	4.008	7.736	15.84
Symbol Rate /(Msymbol/s)	2.004	1.944	2.64
Maximum Processing Time/us	155.11	189.41	243.31

4 Real-time traffic classification

4.1 SVM based multi-class traffic classification

A support vector machine algorithm constructs a hyperplane or set of hyper planes in a high dimensional space, which can be used for classification, regression or other tasks. SVM has ability to simultaneously minimize the empirical classification error and maximize the geometric margin classification space. These properties reduce the structural risk of over-learning with limited samples. There are two key factors while using SVM to classify the traffic. First one is the kernel function which maps the input feature vector to the high dimension hyperplane. In this work we choose the RBF as kernel function. Another one is the regularization parameter C, which can represent the degree of punishment and have great influence on the experiment result.

We start the introduction of traffic classification with the simplest case: classify two traffic classes. Given a training vectors $\mathbf{x}_i \in \mathcal{R}^n, i = 1, 2, \ldots, l$, in two classes, and an indicator vector $y \in \mathcal{R}^l$ such that $y_i \in \{1, -1\}$. Then the optimal classify boundary is solved as the following convex quadratic programming problem:

Table 1 Processing model parameters in us

z_i	c	$u_s(z_i, r_i)$	
		a	b
25	23.8	4.9	24.4
50	41.98	6.3	70

$$\begin{aligned} &\min_{\mathbf{w},b,\xi} \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{l}\xi_i \\ &s.t. \quad y_i(\mathbf{w}^T\phi(\mathbf{x}_i) + b) \geq 1 - \xi_i \\ &\qquad \xi_i \geq 0, i = 1, 2, \ldots, l \end{aligned} \tag{4}$$

where $\phi(\mathbf{x}_i)$ maps $\mathbf{x}_i$ into a higher-dimensional space and $C > 0$ is the regularization parameter. This primal optimization problem could be transferred to the following dual problem, which could lower the computing complexity due to the possible high dimensionality of the vector variable $\mathbf{w}$:

$$\begin{aligned} &\min_{\alpha} \frac{1}{2}\alpha^T\mathbf{Q}\alpha - e^T\alpha \\ &s.t. \quad y^T\alpha = 0 \\ &\qquad 0 \leq \alpha_i \leq C, i = 1, 2, \ldots, l \end{aligned} \tag{5}$$

where $e = [1, \ldots, 1]^T$ is the vector of all ones, $\mathbf{Q}$ is an l by l positive semidefinite matrix, $\mathbf{Q} \equiv y_iy_jK(x_i, x_j)$, and $K(x_i, x_j) \equiv \phi(x_i)^T\phi(x_j)$ is the kernel function. Here, we choose the widely used RBF kernel $e^{-\gamma\|x_i - x_j\|^2}$ to evade prohibitive computation cost while computing $\phi(x_i)$. Notice that, both of C in Eq. (3) and γ are user specified parameters, which has significant influence over the classification results. Hence in the next subsection, we will discuss on how to optimize them.

After solving the Eq.(4), using the primal-dual relationship, the optimal $\mathbf{w}$ satisfies

$$\mathbf{w} = \sum_{i=1}^{l} y_i\alpha_i\phi(x_i) \tag{6}$$

Then the decision function is

$$sgn\left(\mathbf{w}^T\phi(x) + b\right) = sgn\left(\sum_{i=1}^{l} y_i\alpha_iK(x_i, x) + b\right) \tag{7}$$

Hence if Eq. (6) is positive, then the $\mathbf{x}_i$ is classified as $y_i = 1$; otherwise, $\mathbf{x}_i$ is classified as $y_i = -1$. We store $y_i\alpha_i$,

b, $\mathbf{x}$ and other information such as C and γ in the model for prediction.

To extend the two traffic class classification to multi-class one, we use the so-called "one-against-one" approach. Suppose there are k classes, then $k(k-1)/2$ classifiers are constructed and each one trains data from two classes. For training data from the ith and the jth classes, we solve the following two-class classification problem.

$$\begin{aligned} &\min_{\mathbf{w}^{ij},b^{ij},\xi^{ij}} \frac{1}{2}\left(\mathbf{w}^{ij}\right)^T\mathbf{w}^{ij} + C\sum_k(\xi_{ij})_k \\ &s.t.\left(\mathbf{w}^{ij}\right)^T\phi(\mathbf{x}_k) + b^{ij} \geq 1 - \xi_k^{ij}, \quad \text{if } x_k \text{ is in the } i\text{th class} \\ &\left(\mathbf{w}^{ij}\right)^T\phi(\mathbf{x}_k) + b^{ij} \leq -1 + \xi_k^{ij}, \quad \text{if } x_k\text{is in the } j\text{th class} \end{aligned} \tag{8}$$

The voting strategy is deployed for the classification, each binary classification is considered to be a voting, in the end a point is classified to be in a class with the maximum number of votes.

4.2 Parameter tunning

For SVMs, in particular kernelized SVMs, the parameter tunning is crucial but non-trivial. As mentioned, the key parameters of the SVM with RBF kernel function is the soft margin C and the regulationary parameter γ. In this work, we use the method of grid searching, combined with cross-validation to obtain the optimal combination of (C,γ).

Grid search algorithm is a kind of exhaustive search method. Thanks to the independent of C and γ, we could built a grid that set C and γ as the horizontal and vertical axes. The research region of C and γ is $\left[2^{c\min}, 2^{c\max}\right]$ and $\left[2^{g\min}, 2^{g\max}\right]$, respectively. We set the searching step size is 1 for both parameters, then the optimal parameter combination could be found by searching all grid points of (C,γ). The cross validation (CV) is performed for each (C,γ), the (C,γ) that with the highest CV is the optimal parameters. We then use the optimal parameters to train the whole training set and generate the final model. In this work, we set the value of $c\min$ and $g\min$ as -8, and the value of $c\max$ and $g\max$ as 8. Parameter C controls the largest hyperplane and minimizes the data point deviation. After deploying the grid searching and cross validation over the data set, we obtain the value of C is 32768 and γ is 8, and the CV accuracy is 79.9729 percent. The grid searching algorithm is demonstrated in Algorithm 1.

Algorithm 1 Parameter Optimization Algorithm

```
$c = \gamma = 2^{-8}$,$m = 0$;
while $C < 2^8$ do
    $C = 2^{-8} + m$,$m = m + 1$,$n = 0$;
    while $\gamma < 2^8$ do
        $\gamma = 2^{-8} + n$,$n = n + 1$;
        Use the current $\gamma$ and C for classification.Calculate and record classification accuracy,C and $\gamma$.
    end while
end while
Sorting the classification accuracy, return $C$ and $\gamma$ with the highest CV accuracy.
```

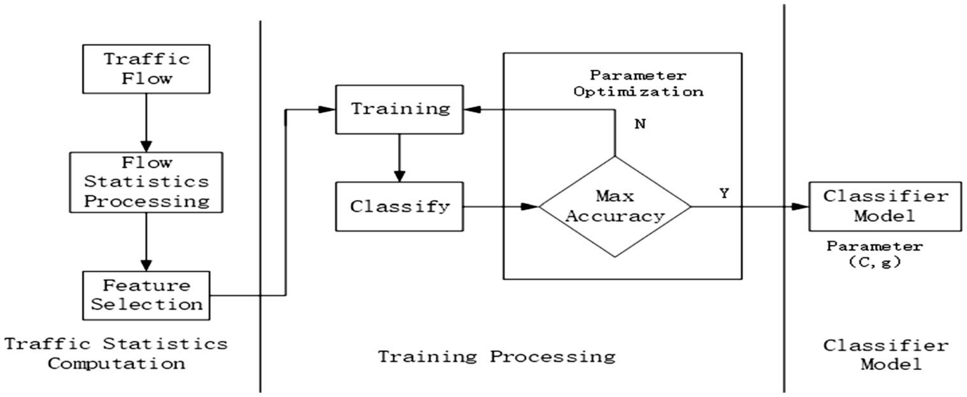
4.3 Classifier

Figure 4 captures the overall training and testing process that results in a classification model. In this work, we train the classifier by providing two kinds of IP traffic: traffic matching the class of traffic that we wish later to identify in the network (in this case the web surfing, video stream, QQ and E-mail) and representative traffic of entirely different applications we may see in the future. The two traffic set are labeled before the training. As shown in the figure, first a mix of traffic traces are collected that contain both instances of the application of interest and instances of other interfering applications (such as DNS, SSH and Peer2Peer file sharing). The "flow statistics processing" step involves calculating the statistical properties of these flows (such as mean and variance of the inter-arrival time, packet length, etc,.) as a prelude to generating features. The following step is selecting the features of captured packets. As we will observe later, different features will cause obviously accuracy and computation complexity gap. The feature selecting is also carried out in cross-validation manner. Different features of two-thirds of the labeled data set are selected for training, and the remaining one-third of the data are used to verify the classification accuracy. The aforementioned parameter optimization step is also implemented for each cross-validation procedure. The features that obtain the highest accuracy are used to classify the traffic later, and the associated parameters are adopted as the final parameters.

We capture the packet from the servers of our campus, which is equipped with Intel E5-2600V4\DDR4 64G\ Broadcom NetXtreme Gigabit Ethernet. The experimental environment is shown in Fig. 5. We use wireshark to monitor the server's port and store the packets' information.

We extract 10 kinds of features from the data packets which is shown in Table 3. However, UDP flows do not

Fig. 4 Training procedures



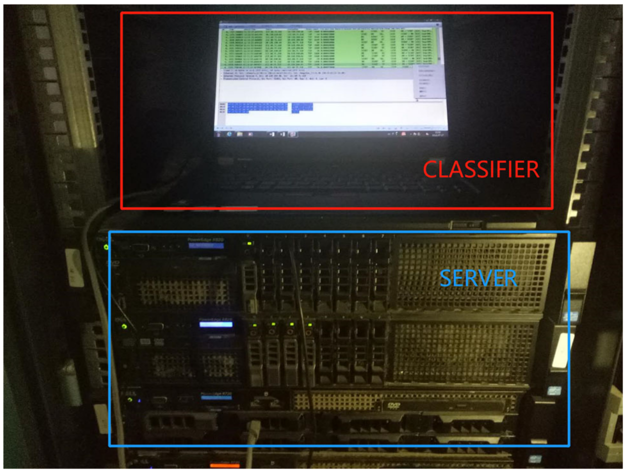


Fig. 5 Experimental environments

have window size, so it can not be used to classify. In addition, according to other articles and our previous work, we chose 1, 2, 3 and 4 as our classification features. In the same time, we label each feature to be used later.

While selecting the directly captured packet features as the classification features, we also take the mean and variance of the packet length and interval time as the classification features. Among them, we choose to have the same protocol, source address, destination address, source port and destination port to the same stream, and the mean and variance is calculated for same stream. Eight types of features shown in the Table 4 are studied for feature selection.

In each cross-validation test, we use the traffic traces which only include the traffic of interest. Since most of the data packets that are initially captured are control information (i.e. SYN, ACK), and the packet length is relatively small which cannot show the different between the classes, hence we ignore the packets of this part and select the data packets of the main body.

We divided into the following combinations to classify the packets and observe their classification accuracy. As shown in Table 5, we divide the feature combination into 7 types. We train each feature combination and then classify by the classifier. We chose the first two-thirds of the packet for training, and the second one-third is classified. For each type of data, we also conduct the classification test separately.

The classification results are shown in the Fig. 6. From the figure, we can see that the classification accuracy different feature combination is different.At the same time,-considering [27] mentioned that the source port number and the destination port number are not suitable for classifying some encrypted application, so we finally chose the F class as the data feature of our classification.

4.4 Real-time traffic classification

As shown in Fig. 7, when a real-time traffic flow arrives,the mean and variance of the packet of the real-time

Table 3 Captured data characteristics

1	Packet length	6	Arrival time
2	Interval time	7	Protocal
3	Source port	8	Source address
4	Destination port	9	Destination address
5	Window size	10	Stream index

Table 4 Selected data characteristics

1	Packet length	5	Mean packet length
2	Interval time	6	Variance packet length
3	Source port	7	Mean interval time
4	Destination port	8	Variance interval time

Table 5 feature combination

A	1,2	E	1,2,3,4,7,8
B	3,4	F	3,4,5,6,7,8
C	1,2,3,4	G	1,2,3,4,5,6,7,8
D	1,2,3,4,5,6		

Table 6 The period of four types tasks

Type	OICQ	Video	E-mail	Browse the web
Period/us	0.009203	0.002664	0.003743	0.025081

traffic flow are calculated. Next,the features of the packet input into the classifier for classification. In this way,the packets are classified into a specific class. The period, processing time,packet length,arrival time ,interval time and deadline of each data packet can be get according to the result of the classification. Among them, The period, processing time,and deadline can be computed by the formula in Sect. 3, and the the packet length,arrival time and interval time can be read in the head of the packet. Then the data packet is scheduled according the features. The period CT_p of each traffic class is given in Table 6.

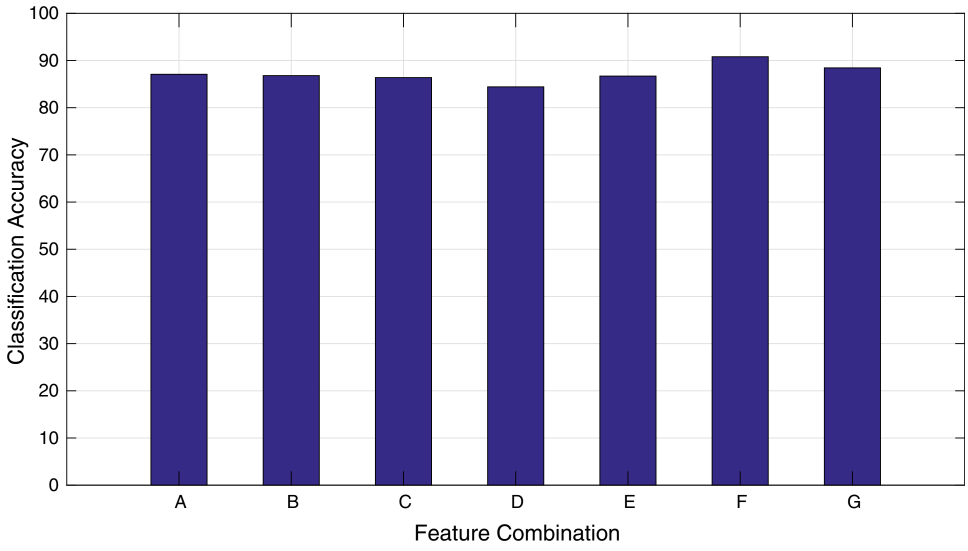


Fig. 6 Classification accuracy of different feature combination

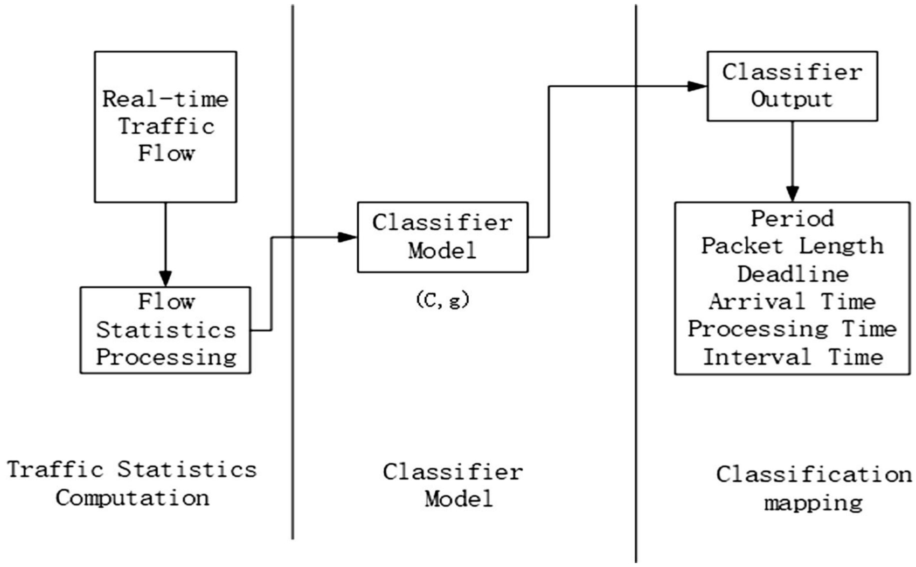


Fig. 7 Classification of real-time traffic flow

5 Scheduling

5.1 Preemptive EDF scheduling

Among 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. As shown in Fig. 8, EDF algorithm is a typical representative of the dynamic priority scheduling algorithm, which is more flexible to meet the QoS requirement of the flows.

Preemptive EDF scheduling algorithm always schedule the packet with earliest deadline, which is deployed under 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.

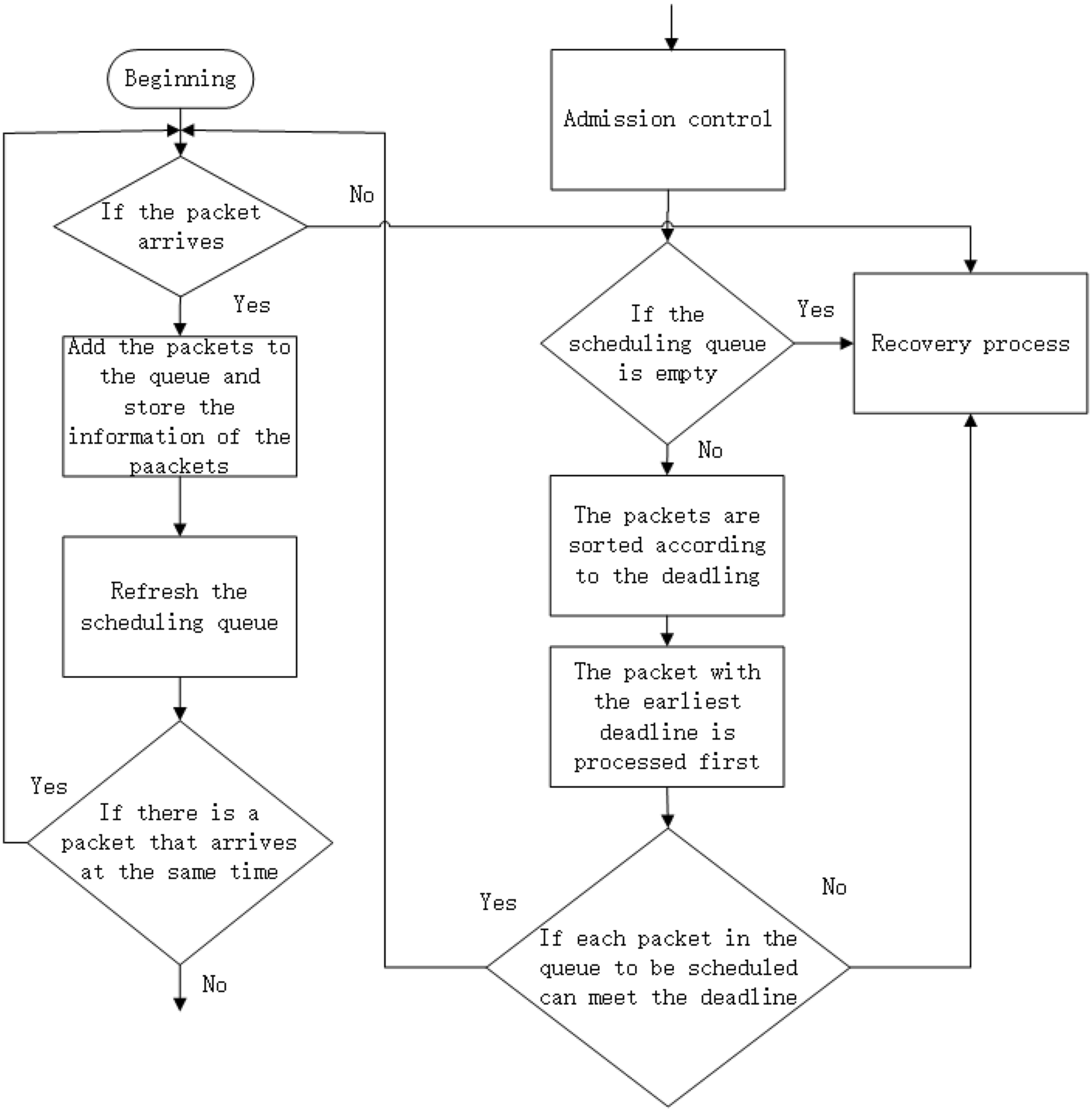

Beginning
If the packet arrives
No
Yes
Add the packets to the queue and store the information of the paackets
Refresh the scheduling queue
Yes
If there is a packet that arrives at the same time
No
Admission control
If the scheduling queue is empty
Yes
Recovery process
No
The packets are sorted according to the deadling
The packet with the earliest deadline is processed first
Yes
No
If each packet in the queue to be scheduled can meet the deadline

Fig. 8 Process of EDF

In this work we adopt preemptive EDF as the scheduling algorithm, and set the scheduling cycle to be 1 us. At the beginning of each scheduling cycle, the arrival of the data packet is monitored. If not, the idling or data recovery operation is performed. If yes, the data packet is added to the queue to be scheduled, and the information of the data packet is stored into the scheduling queue. After the the scheduling queue is refreshed, the scheduler should judge whether the newly arrived packets arrive at the same time. If there is a data packet that arrives at the same time, the queue operation is repeated. Otherwise, the scheduling process is started. The scheduling starts with an admission control which will be discussed later, and after that enters EDF scheduling. While EDF scheduling, the scheduler determines whether the queue is empty. If it does, the current cycle does not need to be scheduled, and idle or data recovery procedure is performed. If there is any data packet in the queue, then the EDF scheduling is performed and the packets are sorted according to the deadline. The packet that with the earliest deadline is processed first. Then, the scheduler checks whether each packet in the queue to be scheduled can meet the deadline, and directly discard the mismatch packets, and then the recovery process is performed by the admission control. After all above operations, the scheduler go to the next schedule cycle.

5.2 Computing aware admission control

According to the EDF algorithm, the most important feature is to predict in advance whether the scheduler can perform in the current working environment and adjust the scheduler scheduling according to the predicted result to ensure that the scheduler can perform normally and reduce the packet loss.

When users surf online which generates a series of work, the generated data packets arrive in real time. We divide the data packets into four categories according to the extracted features, and perform EDF-based scheduling on these four classes to improve CPU efficiency, but in the past The EDF scheduling algorithm will directly drop a part of the data packet that exceeds the CPU performance when the CPU utilization is full.The feature of the packet is show in Fig. 9. This is a great loss to the user experience and this reduces the service efficiency of the network when it is busy. For such reasons, we propose an admission control method to improve the working capacity of the CPU and reduce the number of CPU lost packets to improve the network service quality.

The core idea of the admission control involved is to predict the scheduling process that the scheduler is about to face and make a series of initial judgments under such circumstances. In the current network environment, no one can guarantee when the data packet arrives, so in order to cope with the real-time changing network environment, the scheduler must perform admission control before each scheduling, although this requires higher on CPU.The process of the admission control is shown in Fig. 10.

The process of prediction is to calculate whether the scheduler is running at full capacity for the next period of time. Whenever a packet arrives, a prediction can save the computational resources occupied by the prediction and if no new packets arrive, the predicted result will not change. Firstly, find out the deadlines of all the packets in the sequence to be scheduled, sort by size, for the deadline of the pth data packet, denote DT_p, and its calculation time is ST_p, we have the formula:

$$ST_{psum} = \sum_{n=1}^{p} ST_n \tag{9}$$

According to the comparison between the obtained ST_{psum} and DT_p, if DT_p is smaller, it means that the scheduler

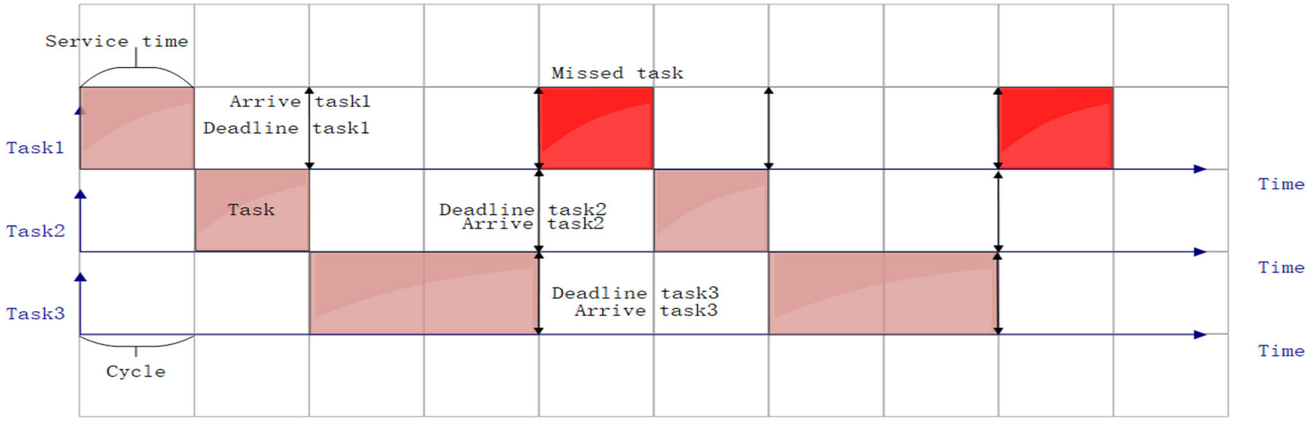


Fig. 9 The parameters of admission control

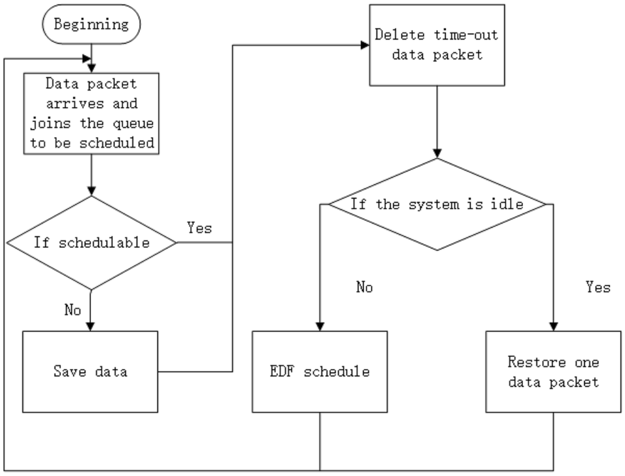


Fig. 10 The process of admission control

cannot complete the task of the data packet until the time to DT_p, and then:

$$ST_{save} = ST_{psum} - DT_p \tag{10}$$

The obtained ST_{save} is the saved data part that cannot be completed. When the scheduler has idle, the data is restored for processing. Of course, the premise is to meet the QoS requirements of the data packet. The rest of the situation indicates that the task of the packet can be scheduled.Then there is the process of data processing. The processing process consists of two parts. The first part is to store the excess part of the overloaded data packet, and the second part is to process the previously stored data packet when the system has no work in the current cycle.

The processing is executed when the system is idle for the cost of time. When saving the data packet, judge all the data packets in the queue to be scheduled. According to the deadline of the data packet, for each deadline, calculate the total service time of the data packet before the deadline. If the total service time exceeds the deadline, then the process cannot be scheduled by the system, and the difference between the deadline and the service time is saved as the service time for saving the data packet.

The number of storage spaces is related to the number of classifications of the classifier. Different types of data packets are stored in different storage spaces. When the data packet is restored and used, the recovery order is specified according to the requirements of the quality of service, that is, the order of the storage space. When there is a chance to recover, traverse in order.

Algorithm 2 Admission Control

```
Input: matrixes-process, now-time;
Output: flag, now-time;
for all $i$ <=matrixes-process-length do
  if Now-time+$C_{isum} > D_i$ then
    Savedata=now-time+$C_{isum}$-$D_i$;
    if $T_i$ =data-packet-type[k] then
      space-save-data[k]=Savedata
      num-save-data++
    end if
    $C_{isum} = D_i$−now-time
  end if
end for
if num-save-data>0 then
  if num-save-data-type[k]>0 then
    while i<num-save-data-type[k] do
      if deadline-save-data[i]>now-time then
        Delete save-data[i]
      end if
      i++
    end while
  end if
end if
if matrixes-process-length= 0 then
  if num-save-data-total> 0 then
    if num-save-data-type[k]> 0 then
      save-data=save-data-time-unit
      now-time=now-time+time-unit
      if save-data<=0 then
        Delete save-data
      end if
    end if
  end if
end if
```

The second part is the work of restoring the save. Since the CPU has no time to work, the work that has been squeezed out is saved by calculation. After saving the work, it is detected whether the current time unit of the CPU will be occupied, whether there is idle time, if not, then enter the normal scheduling process, but if there is, you can use this time to restore the previously saved business and the packet discarding, determining whether there is a saved data packet by the size of each storage space, and if so, first performing a packet loss processing judgment, that is, when the current time is greater than the deadline for storing the data packet, discarding the data packet, traversing the entire saved data space. Then, the operation of restoring the data packet is performed, and the saved data packet is searched from the set priority order. If the processing is found and the data packet is processed,

Table 7 The influence of parametric optimization on classification accuracy

Type	OICQ	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

Table 8 The time and accuracy to train different feature combinations

Feature combination	Time/s	Accuracy	Accuracy/time
A	499	0.8708	0.001745
B	299	0.868	0.002903
C	704	0.8638	0.001226
D	794	0.8442	0.001063
E	757	0.867	0.001145
F	238	0.908	0.003815
G	852	0.8844	0.001038

and if the processing is completed, the saved data is deleted, otherwise the operation is not performed. Only one packet is processed at a time, and the processing returns to the schedule after the processing ends.

There are two levels of sequence here. The first layer is the sequence of multiple classification result storage spaces. This order is different according to the different order of service quality requirements, and the second is the sequence of arrival of data packets in the space. The order of these two layers determines the recovery order of work.

6 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 7. These tasks is classified by SVM algorithm which is described in Sect. 4.

For different class of traffic, the SVM with optimization of the parameter has different effection.From the Table 7, we can see clearly that using parameter optimization can improve the accuracy of classification.

As shown in the Table 8, for different feature combination,the training time of classification is different,and the accuracy of classifying the same data set is different too. We use the ratio of accuracy and time to characterize the efficiency of the training. It can be seen from the Table 8 that the combination of features of class F has higher efficiency, which can ensure higher classification accuracy and shorter training time.

The classifier is trained by the same set of packets,and the classifier is used to classify the feature combination of the same class,which increases as the number of data in the packets increase,but when the number of packet is large enough,the difference of time is small.However, the time is independent of the number of features in the feature combination when the classifier performs classification.When the combination of the B and F features combines the same number of data sets, the time required is basically the same, but the accuracy of the classification is different. Feature combination G require the most time to combine classification data.It can be seen from the Fig. 11 that the features that have a greater impact on the classification time are the packet length and the interval time.The mean and variance of the packet length and the interval time have less influence on the classification time.

The total processing time can be modelled as a function of CPU frequency.According to the relationship between CPU frequency and processing time, we analyzed the impact of admission control on packet loss rate, delay and throughput under different CPU frequencies.

The combination of classification and scheduling could not only consider the diverse QoS requirement of different

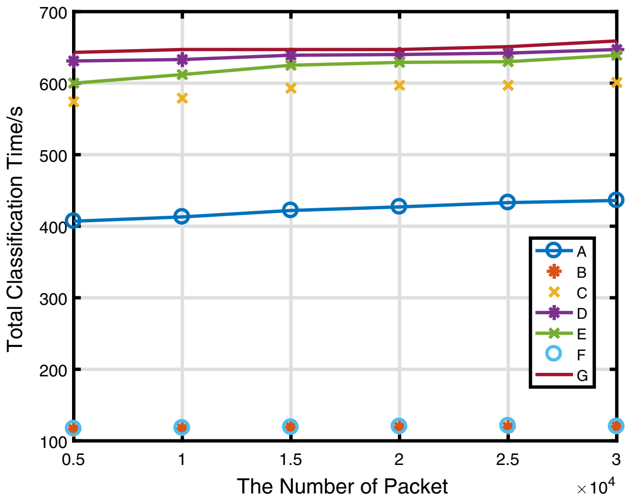


Fig. 11 The influence of different feature combinations on time

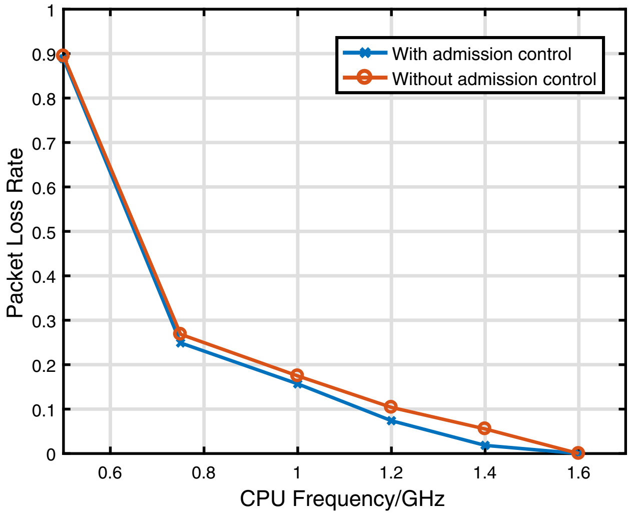


Fig. 12 The influence of different frequency on packet loss rate

traffic class, but also consider the baseband signal processing load. Moreover, it can obtain lower packet loss rate, lower computing complexity and higher computing resource utilization.

It can be seen from Fig. 12 that the system frequency has a large packet loss rate between 0 and 0.8G, and the use of admission control shows a trend of getting better and then getting worse between 0.8G and 1.6G, and is equal after 1.6G. Even in the case of 0, the possible reasons are as follows:

When the system frequency is low, it can't be processed at all or it can't handle the arrival of network packets, the system will have an unschedulable state. In this state, there is no idle time left to the admission control for recovering the data packet, resulting in the loss. The case where the packet rates are almost equal.

In order to meet the requirements of low latency, the data packet storage time is limited, and it will be lost when the deadline is reached. If the system cannot be idle, the admission control has no chance to recover the data packet. In this case, admission control can not improve the system's condition, but will increase the system's computational overhead and reduce the efficiency of the system. So in this case, the admission control is not recommended.

When the system frequency is high, the packet loss rate is also nearly the same. Even when the system frequency exceeds a certain value, the packet loss rate will be zero. This situation arises because the system computing power is fully capable of handling the incoming data stream, so in this case, the admission control does not appear to improve the system. In this case, consider using the admission control to cope with the possible sudden flow situation.

When the system frequency is at an intermediate value, the difference in packet loss rate is somewhat reflected. At this time the system will be busy, but not always busy. Sometimes the system will be idle, then you can call admission control. When a burst of data flows, the processing speed of the system cannot completely process the incoming data packet, and the admission control saves the unprocessable part. When the system is processed, if the delay is not exceeded, Recover packets that were not processed before. Therefore, at the intermediate frequency, the packet loss rate curve shows a situation of getting better and then getting worse. In this case, using the admission control can effectively improve the performance of the system.

As can be seen from Fig. 13, the delay aspect tends to decrease as the frequency increases and the delay decreases. However, admission control does not significantly reduce the delay, and sometimes there is an increasing trend. The reason is that the data packet recovered by the admission control is not scheduled by the system and will be performed when the system is idle. At this time, the delay of the data packet is larger than those of the normal processing, and the delay is slightly increased, so there is some performance degradation in terms of delay.

As can be seen from Fig. 14, the throughput is very high at 0.8 Ghz, showing an abnormal trend, but after 0.8G, it will be found that the throughput increases as the frequency increases. The reason for the low frequency throughput is that when the system frequency is too low, the system cannot handle the arrival of these data packets, most of the data packets are discarded, and only a small number of data packets are processed normally. This caused an abnormal situation in which the number of processed data packets was small and the time was also short. When the system frequency increases, it can be seen that the throughput is proportional to the system frequency. When the system frequency increases, the ability to process data packets is

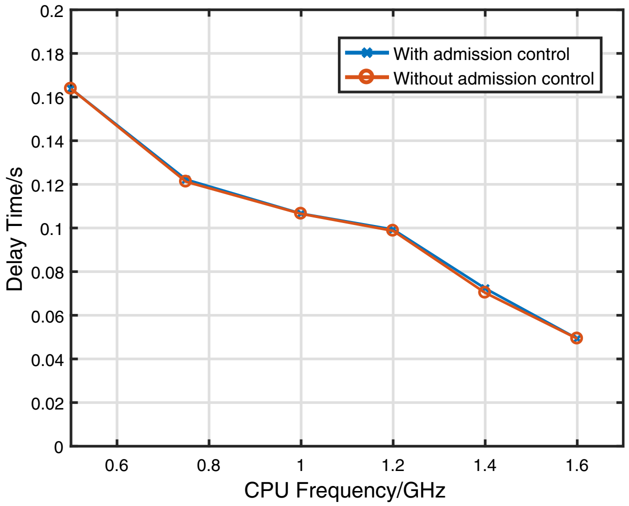


Fig. 13 The influence of different frequency on delay time

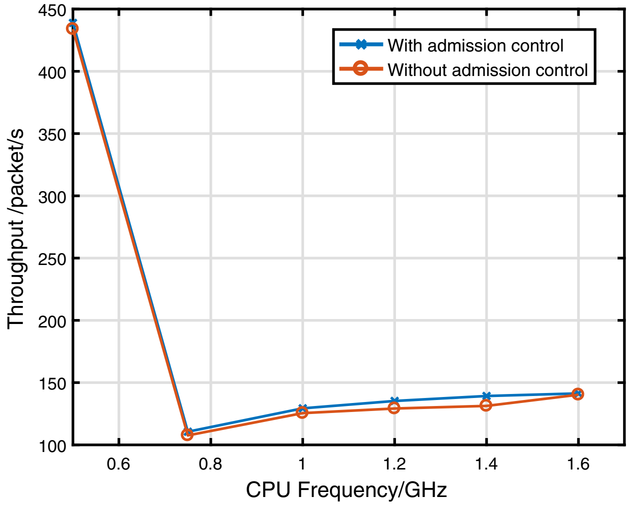


Fig. 14 The influence of different frequency on throughput

also enhanced accordingly. The data packets processed per unit time increase, and the throughput is naturally high. After the admission control is added, the admission control performs data recovery operations in the appropriate frequency range, improving throughput. However, the frequency improvement effect is reduced because the processing power of the system is improved, and the situation in which scheduling is impossible is reduced.

It can be seen from the experimental results that it is crucial that the admission control can effectively predict the situation where the packet will be lost and save it. Although there are few opportunities in the experiment to find the system idle state to restore the previously saved data services, we can see that high system frequency and low system frequency are not ideal results. When the system frequency is low, the number of queues can be seen. Too large, slow processing speed, low throughput, and insufficient ability to handle business. However, when the system frequency is high, the queue has a lot of idleness, and a large amount of computing resources are idle, which greatly reduces the utilization efficiency of the system, but the throughput is high, the delay meets the requirements, there is no packet loss, and the processing speed is fast.

As can be seen from the above, the delay of various frequencies meets the requirements. But for low frequencies, a lot of packet loss and system congestion become the norm, affecting network circulation and user experience. For high frequencies, a large amount of idle time wastes the computing resources of the system and reduces the utilization of the network.

It can be obtained that in the business where the network service is not busy, a lower frequency CPU can be used. High-performance CPUs are used in nodes where network traffic is busy. Admission control is better for systems with lower system frequencies and relatively busy network services. When the system frequency corresponds to the busyness of network services, the adoption of admission control can effectively reduce the packet loss rate and reduce the occurrence of bad phenomena such as user jams. But increase the network delay.

The curve in the figure is in line with expectations. For the case of sudden increase in network traffic, admission control can effectively improve the utilization rate of the system and the service effect. However, for the long-term busy situation of the network service, if the CPU does not have a suitable frequency, the effect of the admission control is not obvious, as shown by the curves in the figure.

In this work, we study the performance of EDF with admission control by simulation. For comparison, the scheme adopted by LTE is used as a benchmark. To decrease the complexity order, we proposed LTE which is a simplified algorithm. Specifically speaking, the CPU deploys a fixed priority scheduling, where the server depends on the predefined priority as defined by Table 6.1.7 in specification [3].

In Fig. 15, we present the variations of processing time obtained by the EDF algorithm and LTE versus the number of packet. We can observe that EDF algorithm with admission control have lower processing time than LTE, and the performance gap is increasing with the number of packet. It can validate the efficiency of EDF algorithm with admission control.

In Fig. 16, we show the variations of packet loss rate obtained by the EDF with admission control and LTE versus the number of packet. It is obvious that EDF algorithm has much better performance than LTE. We can see that the packet loss rate basically stay the same when the number of packet reaches 2000 under EDF, while the packet loss rate closes to 1 under LTE.

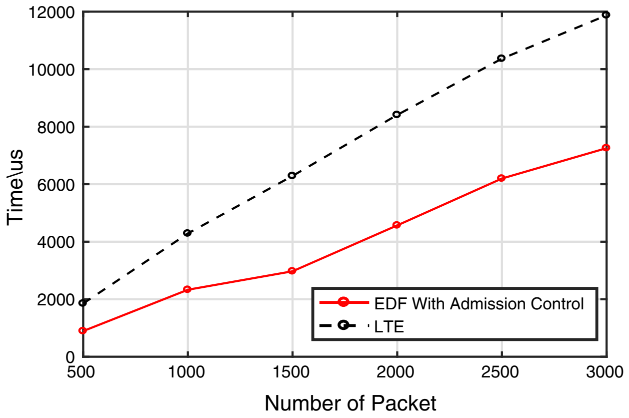


Fig. 15 Processing time versus of number of packet

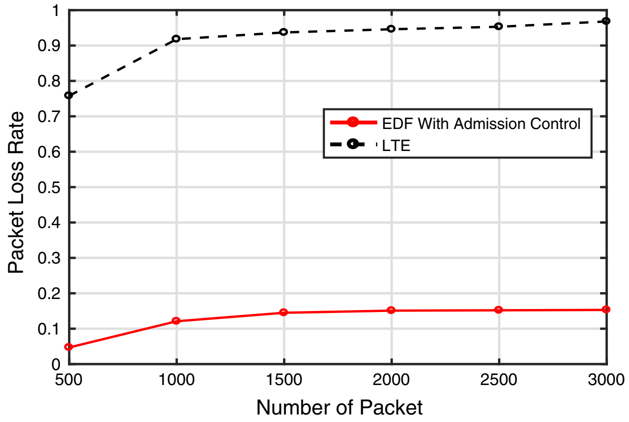


Fig. 16 Packet loss rate versus of number of packet

7 Conclusion

In this work, we have proposed a computing aware scheduling algorithm in MEC. The proposed scheduling algorithm reinforce the traditional LTE scheduler by combining the real-time traffic classification and CPU scheduling. The SVM based traffic classification algorithm with parameter tunning and training procedures are given, best combination of features is obtained based on extensive cross-validations. The preemptive EDF scheduling with admission control is given in the second place. It attempts to break the barriers between the computation and wireless communications. Numerical results have illustrated the efficiency of the preemptive EDF scheduling with admission control, which showing advantage over the preemptive EDF scheduling without admission control and LTE.

The classification method in this study is supervised learning mode, and the focus of our future work is to study and evaluate other classification methods in unsupervised learning mode. Meanwhile, it's also an important future direction how to re-train classifiers and extend the life of classifiers.

Acknowledgements This work is jointly supported by Project 61501052 and 61602538 of the National Natural Science Foundation of China, and Project D010109.

References

1. Mao, Y., You, C., Zhang, J., Huang, K., & Letaief, K. B. (2017). A survey on mobile edge computing: The communication perspective. *IEEE Communications Surveys & Tutorials*, *19*(4), 2322–2358.
2. Medium access control (MAC) protocol specification, 3GPP Std. TS 36.321, Sep. 15.3.0 (2018).
3. Policy and charging control architecutre, 3GPP Std. TS 23.203, Sep. 15.4.0 (2018).
4. Capozzi, F., Piro, G., Grieco, L. A., Boggia, G., & Camarda, P. (2013). Downlink packet scheduling in lte cellular networks: Key design issues and a survey. *IEEE Communications Surveys & Tutorials*, *15*(2), 678–700.
5. Wanstedt, S. (2007). Mixed traffic hsdpa scheduling-impact on voip capacity. In *Vehicular Technology Conference (2007). VTC2007-Spring. IEEE 65th*. IEEE (pp. 1282–1286).
6. Shakkottai, S., & Stolyar, A . L. (2002). Scheduling for multiple flows sharing a time-varying channel: The exponential rule. *Translations of the American Mathematical Society-Series 2*, *207*, 185–202.
7. Wang, K., Yang, K., Chen, H. H., & Zhang, L. (2017). Computation diversity in emerging networking paradigms. *IEEE Wireless Communications*, *24*(1), 88–94.
8. Nikaein, N. (2015). 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 (pp. 36–43).
9. Valenti, M. C., Talarico, S., & Rost, P. (2014). The role of computational outage in dense cloud-based centralized radio access networks. In *2014 IEEE Global Communications Conference*. IEEE (pp. 1466–1472).
10. Rost, P., Maeder, A., Valenti, M. C., & Talarico, S. (2015). Computationally aware sum-rate optimal scheduling for centralized radio access networks. In *2015 IEEE Global Communications Conference (GLOBECOM)*. IEEE (pp. 1–6).
11. Guo, K., & Sheng, M. (2016). Cooperative transmission meets computation provisioning in downlink c-ran. In *2016 IEEE International Conference on Communications (ICC)*. IEEE (pp. 1–6).
12. Ha, V. N., & Le, L. B. (2016). Computation capacity constrained joint transmission design for c-rans. In *2016 IEEE Wireless Communications and Networking Conference*. IEEE (pp. 1–6).
13. Liao, Y., Song, L., Li, Y., & Zhang, Y. A. (2016). Radio resource management for cloud-ran networks with computing capability constraints. In *2016 IEEE International Conference on Communications (ICC)*. IEEE (pp. 1–6).
14. Molina Pena, M., Muñoz Medina, O., Pascual Iserte, A., & Vidal Manzano, J. (2014). Joint scheduling of communication and computation resources in multiuser wireless application offloading. In *Proceedings PIMRC 2014*. Institute of Electrical and Electronics Engineers (IEEE) (pp. 1093–1098).
15. Yu, Y., Zhang, J., & Letaief, K. B. (2016). Joint subcarrier and cpu time allocation for mobile edge computing. In *Global Communications Conference (GLOBECOM), 2016 IEEE*. IEEE (pp. 1–6).
16. Yang, L., Cao, J., Cheng, H., & Ji, Y. (2015). Multi-user computation partitioning for latency sensitive mobile cloud applications. *IEEE Transactions on Computers*, *64*(8), 2253–2266.
17. Jing, N., Yang, M., Cheng, S., Dong, Q., & Xiong, H. (2011). An efficient svm-based method for multi-class network traffic classification. In *Performance Computing and Communications Conference (IPCCC) (2011). IEEE 30th International*. IEEE (pp. 1–8).
18. Hao, S., Hu, J., Liu, S., Song, T., Guo, J., & Liu, S. (2015). Improved svm method for internet traffic classification based on feature weight learning. In *2015 International Conference on Control, Automation and Information Sciences (ICCAIS)*. IEEE (pp. 102–106).
19. Yamansavascilar, B., Guvensan, M. A., Yavuz, A. G., & Karsligil, M. E. (2017). Application identification via network traffic classification. In *2017 International Conference on Computing, Networking and Communications (ICNC)*. IEEE (pp. 843–848).
20. Li, Z., Yuan, R. , & Guan, X. (2007). Accurate classification of the internet traffic based on the svm method. In *IEEE International Conference on Communications, ICC'07*. IEEE (pp. 1373–1378).

21. Bhaumik, S., Chandrabose, S. P., Jataprolu, M. K., Kumar, G., Muralidhar, A., Polakos, P., Srinivasan, V., & Woo, T. (2012). Cloudiq: A framework for processing base stations in a data center. In *Proceedings of the 18th annual international conference on Mobile computing and networking*. ACM (pp. 125–136).
22. Liu, C. L., & Layland, J. W. (1973). Scheduling algorithms for multiprogramming in a hard-real-time environment. *Journal of the ACM*, *20*(1), 46–61.
23. Bastoni, A., Brandenburg, B. B., & Anderson, J. H. (2010). An empirical comparison of global, partitioned, and clustered multiprocessor edf schedulers. In *Real-Time Systems Symposium (RTSS) (2010). IEEE 31st*. IEEE (pp. 14–24).
24. Sesia, S., Toufik, I., & Baker, M. (2009). *LTE, the UMTS long term evolution: From theory to practice*. New York: Wiley.
25. Wang, K., & Cen, Y. (2017). Real-time partitioned scheduling in cloud-ran with hard deadline constraint. In *2017 IEEE Wireless Communications and Networking Conference (WCNC)* (pp. 1–6).
26. Physical layer procedures (fdd), 3GPP Std. TS 36.213, Sep. 15.3.0 (2018).
27. Nguyen, T. T. T., & Armitage, G. (2009). A survey of techniques for internet traffic classification using machine learning. *IEEE*, *10*(4), 56–76.

Ke Wang received his Ph.D. degree in communication and information system from Beijing University of Posts & Telecommunications (BUPT), China, in 2014. Now he is an assistant professor with the School of Information and Communication Engineering in BUPT. From 2017, he temporary transfers to Ministry of Industry and Information Technology of China (MIIT) as a technical coordinator of the National Science and Technology Major Project “New Generation Broadband Wireless Mobile Communication Network”. Currently, he has lead two projects under China NSF and CASC funding for the next generation satellite networks design. He has published more than 50 papers in the area of mobile communications and served several IEEE conferences (ICC/Globecom/WCNC/PIMRC) as TPC members. He is also a web-chair of EAI SmartGift 2018.

XiaoYi Yu received the B.Eng. degree in Communication Engineering from Shandong University of Science and Technology in 2017. She is currently pursuing the Ph.D. degree at School of Electronic Engineering, BUPT. Her research interests include satellite networking and hybrid satellite-aerial-terrestrial networks.

WenLiang Lin received the B.S. degree and Ph.D. degree of electronic engineering in 2010 and 2016, from BUPT. He has been a post-doctoral candidate from BUPT since 2016. He has participated in numerous CASC projects and lead the research area of satellite protocol monitoring equipment design from 2012 to 2015. His research interests include design and performance evaluation of satellite mobile communication, satellite channel modeling and traffic steering strategy. He has published 9 journal papers and 9 Chinese patents in this field.

ZhongLiang Deng received the B.S. from Beihang University in 1991, and received the Ph.D. from Tsinghua University in 1994, Beijing, China. Since 1996, he has been working in BUPT as a full professor. He is the leader and principal scientist of the National Key Science and Technology project “XIHE” of Indoor Positioning and Navigation. Also, he is one of the General Experts for the National Key Research and Development Plan Committee. He has been elected as the TOP 10 Scientists of China in 2014 and granted the Chinese highest technical award of engineering “GuangHua Engineering Award” in 2016. He has received 2 National Science and Technology Progress Award of China. He is the fellow of Chinese Institute of Electronics. He has published over 100 Chinese Patents and more than 300 technical papers.

Xin Liu master candidate with School of Information and Communication Engineering, Beijing University of Posts and Telecommunications. His current research interests include wireless networking, cooperative networks and hybrid satellite-aerial-terrestrial networks.