2020 International Conference on Computer, Network, Communication and Information Systems (CNCI 2020)

Dynamic Token Allocation Strategy Based on Weight in TCSN System

Zhe Du[1,a], Xiaowen Dai[1,b] and Jing Yu[1,c]

[1]*School of Electrical Engineering, Southwest Jiaotong University, Chengdu, Sichuan, China*
a. duzhe_a@foxmail.com, b. xwdai678@163.com, c. 1361232395@qq.com

Keywords: Token bucket, traffic shaping, packet loss rate, entropy weight method.

Abstract: In the Train Control and Service Network (TCSN) system, ensuring the quality of train control and passenger services is important. However, traditional traffic shaping methods are static, they can't adapt to the changing state of the network load. Therefore, the dynamic token allocation strategy based on weight is proposed. On the basis of traditional token bucket traffic shaping method, an overflow token bucket is added, and a two-level token allocation algorithm is used to dynamically adjust. Taking the current packet loss rate of each bucket, the number of cached packets and the number of tokens in the bucket as evaluation indicators. So the weight of the token bucket is determined. And combined with the weighted maximum and minimum fair allocation algorithm, tokens in the overflow bucket can be fairly distributed, therefore the packet loss rate of each bucket has dropped greatly.

1. Introduction

Nowadays, the most of trains are still using the Train Control Network(TCN) system [1], which can only transmit the service of the train but the passengers' can't. Thus the train control and service network(TCSN) is proposed, which is an integrated transmission network system and it can transmit both train control information and passenger services [2]. So the experience of the passengers will be greatly improved.

In order to established the reliable TCSN system, the research methods of computer traditional network are referred. So the article borrows some methods of traditional networks and improves these to make them more suitable for TCSN.

There are many traditional network traffic models, such as the Markov model, Poisson process model, and Auto Regressive and Moving Average(ARMA) model [3]. However, traffic sequences generated by these traditional models usually only have a short-time dependence in the time domain, so after averaging over time, their burst characteristics tend to be stable. The short dependence in the time domain corresponds to the high frequency in the frequency domain. Therefore, the sequences generated by these models generally have more high-frequency components, and the relatively dominant ration of low-frequency components is relatively low. Based on current knowledge of network traffic sequence, none of them can accurately describe real network traffic. Although each model has its characteristics, its related structures show an exponential decay, and their mathematical representation is a rapid decay of the auto-correlation function.

For traffic shaping, there are two basic algorithms, which are Leaky Bucket algorithm(LB) and the Token Bucket algorithm(TB).LB can effectively reduce data burst and stabilize the data flow rate of the core network. TB allows bursts of data traffic in a certain size, which can provide a more flexible data flow rate for the core network.

The main work of this article is to establish a network traffic model which simulates the TCSN system, then we improve dynamic token shaping strategy, which can greatly reduce the packet loss rate.

2. Self-Similar Trafffic Model

2.1.Self-similar Model

In recent years, with the deep research of wired and wireless network traffic by a large number of scholars, [4, 5, 6] pointed that most network services have statistical self-similarity.

The most prominent characteristic of self-similarity is bursts of random length and it doesn't change with the time changing. Bursty traffic has long-range dependence(LRD), so it can't easily smooth the bursty traffic. However, traditional models like those based on Poisson or Bernoulli process, such as Markoff Modulated Poisson Process(MMPP), AR process, etc. These models can only effectively deal with short-range dependence(SRD).

As mentioned earlier, the measuring result shows that the traffic in current communication network has LRD. Thus the traditional SRD model can't accurately model and analyze the data traffic change process. Therefore, the self-similar process is used to model the network traffic changes, which can more accurately and realistically simulate the network.

For the TCSN system, we also use a self-similar process to simulate the network.

Assuming that the process of packet arrival is $X = X(t)_{t\geq 0}$ which is a self-similar process. So we can define it as follows:

$$\exists H > 0, \forall c > 0, X(ct) \overset{\text{def}}{=} c^H X(t) \tag{1}$$

In the equation(1), the $\overset{\text{def}}{=}$ represents the same distribution, then X is the H self-similar process, and H is the Hurst parameter which used to measure the degree of self-similarity.

$$\mathrm{Var}\{X(ct)\} = c^{2H}\mathrm{Var}\{X(t)\} \tag{2}$$

the constant c>1, the Hurst parameter $H \in \left(\frac{1}{2}, 1\right]$.

Fraction Brown Motion(FBM) is the most commonly used model in self-similar mathematical models. It is based on the statistical characteristic of long-range dependence to establish the model. Therefore it can be more accurate and flexible to analyze the data.

The definition of FBM is proposed by Norros [7], and details as follows: $Z(t), t \in (-\infty, +\infty), H \in \left(\frac{1}{2}, 1\right]$.

- Z(t) is smoothly increasing;
- $Z(0) = 0, EZ(t) = 0$ *for all* t;
- $EZ(t)^2 = |t|^{2H}$ *for all* t;
- Z(t) is continuous.

● Z(t) is a Gaussian process. For example, its finite-dimensional distribution is a multivariate Gaussian distribution.

Using the FBM to simulate the network traffic $A_i(t), i = 1,\ldots,k$ as follows:

$$A_i(t) = m_i t + \sqrt{m_i a} Z_H(t), t \in R \tag{3}$$

Thus,

$$A(t) = mt + \sqrt{ma} Z_H(t), t \in R \tag{4}$$

In the equation(4), $A(t) = \sum_{i=1}^{k} A_i(t)$, $m = \Sigma_{i=1}^{k} m_i$, and m is the average rate of arriving data traffic. And a is the variance of the coming data traffic. $Z_H(t)$ represents the Gaussian stochastic process, whose mean is 0, and variance is $Var[Z_H(t)] = |t|^{2H}$. The Hurst parameter $H \in \left(\frac{1}{2}, 1\right]$.

3. Token Allocation Strategy Based on Dynamic Weight

3.1.The Improved Token Bucket Model

TB is one of the most commonly used shaping in traffic. As shown in Figure 1, the mechanism of the TB is that the transmission of data packets is controlled by the number of tokens in token bucket.

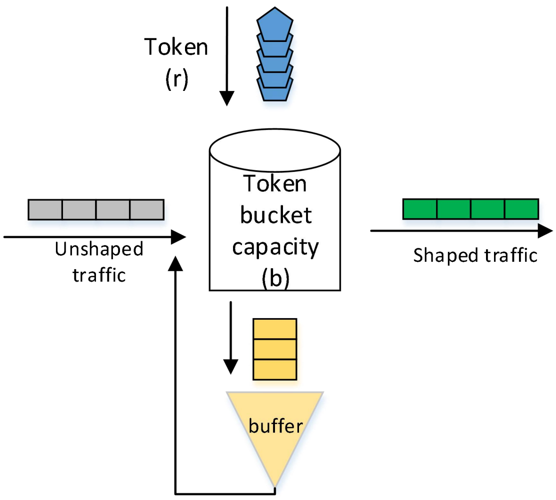

Figure 1: The token bucket algorithm.

There are two parameters in TB. The parameter r is the speed at which token generated, and b is the number of tokens in the bucket. When a packet arrives, it needs to obtain the token before it can be forward. On one hand, when tokens have filled the bucket, the extra tokens generated will be discarded, so some tokens will be wasted. On the other hand, when the token in a bucket is used up, the packet can't be transmitted. What's worse, the packet will be discarded, resulting in a high packet loss rate. Thus, we make some improvements, as shown in Figure 2.

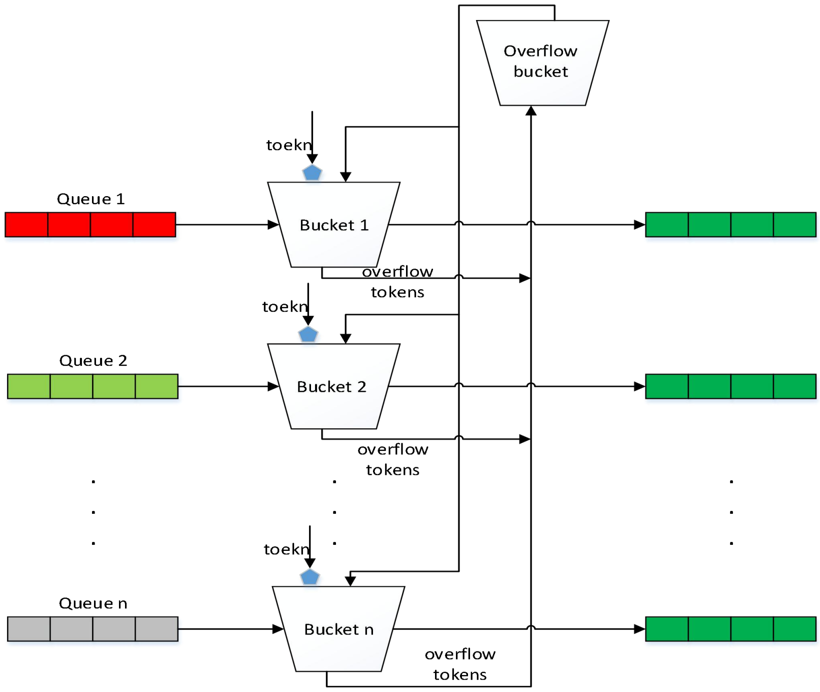

Figure 2: The improved token bucket algorithm.

Assuming that there are n queues of data packets which need to be shaped. Each queue has its own token bucket. However, unlike the initial one, an overflow bucket is added.

The role of the overflow bucket is to store tokens which are discarded by original bucket and redistribute the tokens to the lacking tokens bucket.

3.2.The Token Allocation Weight based on Entropy Weight Method

It can be known from reference [8, 9, 10] that the entropy weight method is an objective method to weighting. It uses the entropy of each index to weight all indicators, so the result of evaluation can be more objective and can be used in many domains.

According to the status of each original token bucket, we allocate the tokens of overflow bucket. Specifically we can take the current number of tokens in the original bucket a_i, and the current number of cached data packets b_i, and the current packet loss rate c_i as evaluation indicators. Based on the entropy weight method, we can determine the token allocation weight for each bucket at current moment.

The steps of entropy weight method are as follows,

3.2.1.Standardizing the Evaluation Index Data

$$Y_{ij} = \frac{X_{ij}-min\,(X_i)}{max\,(X_i)-min\,(X_i)} \text{ or } Y_{ij} = \frac{max\,(X_i)-X_{ij}}{max\,(X_i)-min\,(X_i)} \quad (5)$$

In the equation (5), X_{ij} represents the j_{th} index of the i_{th} token bucket at t time, and Y_{ij} is the standardized data. The former is applied to forward indicators and the latter is applied to reverse indicators.

3.2.2.Calculating the Information Entropy of Each Indicator

$$E_j =-\frac{1}{\ln n}\sum_{i=1}^{n} P_{ij}\ln P_{ij} \qquad (6)$$

In the equation (6), n represents the number of the token bucket. And the definition of P_{ij} is as follows:

$$P_{ij} = \frac{Y_{ij}}{\sum_{i=1}^{n} Y_{ij}} \qquad (7)$$

If $P_{ij} = 0$, then $\lim_{P_{ij} \to 0} P_{ij}\ln P_{ij} = 0$.

3.2.3.Calculating the Weight of Each Indicator

$$W_j = \frac{1-E_j}{k-\sum_{j=1}^{k} E_j} \qquad (8)$$

In the equation (8), k is the number of the indicators. In the article, $k = 3$.

3.2.4.Calculating the Final Score of the Object i（The Article's Object is a Token Bucket）

$$V_i = \sum_{j=1}^{k} x_{ij} W_j \qquad (9)$$

Standardizing the final score data of the toke bucket, we can get the weight of the bucket S_i.

$$S_i = \frac{V_i}{\min V} \qquad (10)$$

In the equation (10), $V = \{V_1,V_2,...,V_i,...,V_n\}$, V is the set of all assigned token weight.

3.3.Fair Token Allocation Algorithm based on Weight

In the above analysis, a dynamic token allocation strategy based on weight is proposed, which can assign a variable weight to each token bucket. According to the status of the token buckets, the weight will be adjusted, so it can maintain the fairness among token buckets in real time.

The key of the allocation strategy is to share the tokens which are belonged to overflow bucket by the weight. It not only satisfies the tokens requirement of each token bucket but also takes the account of fairness among the token buckets, when the tokens of overflow bucket can't satisfy all original buckets. Therefore, this strategy adopts a weighted maximum and minimum fair allocation algorithm.

The weighted maximum and minimum fair allocation algorithm is derived from the improved max_min fair allocation algorithm. It was first adopted to solve the problem of fair allocation of bandwidth. As mentioned in the lecture [11, 12],the algorithm schedules limited resources based on users' weight, and adopts the theory of multiple average to determine the allocation scheme, which can achieve fair distribution in multiple modes. The advantage of using the weighted maximum and minimum fair allocation is as follows:

- By the weight normalization, the resources are allocated in order of increases to demand.
- None can get more resources than it needs.
- Unsatisfied users will continue to share resources by weight.

A flow chart of the token bucket allocation strategy can be obtained, as shown in Figure 3.

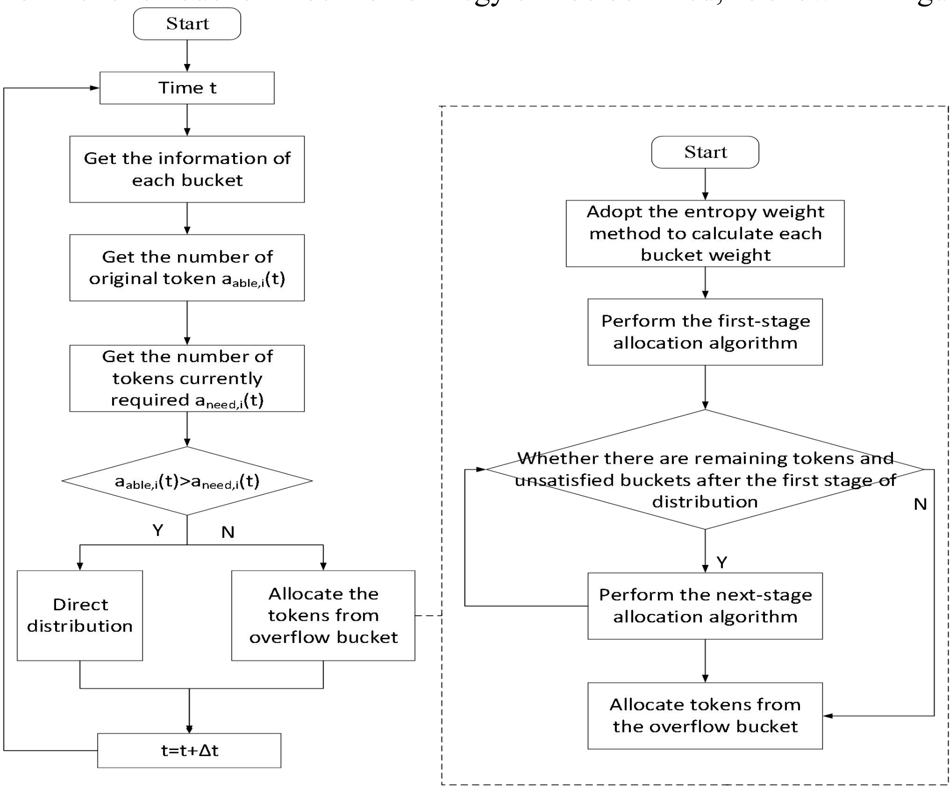

Figure 3: Flow chart of tokens allocation strategy control.

3.3.1. Get Each Token Bucket Status

Get the status information about each token bucket, including the current number of tokens $a_{able,i}(t)$, the number of cached data packets $b_i(t)$, the packet loss rate $c_i(t)$, the number of token in overflow bucket $d(t)$, and the number of tokens that the bucket i_{th} needed $a_{need,i}(t)$.

3.3.2. Compare the $a_{able,i}(t)$ and $a_{need,i}(t)$

If $a_{able,i}(t) \geq a_{need,i}(t)$, it means that the bucket i_{th} is in the stage of oversupply, so it will be allocated directly. Conversely, it means that the bucket I is in the stage of short supply, so the weighted maximum and minimum allocation is performed. The specific allocation process is as follows:

i. Adopt the entropy weight method and using the (10), determine the token allocation weight S_i of each token bucket.

ii. Calculate the number of the tokens that currently required by token bucket i_{th}.

$$a_{req,i}(t) = a_{need,i}(t) - a_{able,i}(t) \tag{11}$$

iii. Implementation the phase 1st allocation, it means that the tokens in the overflow bucket are allocated correspondingly according to the weight calculated previously. So the number of tokens that token bucket i_{th} obtained in phase 1st is as follows:

$$a_i^1 = d(t)\frac{S_i}{\sum S_i} \tag{12}$$

iv. Count the number of tokens that is allocated by the overflow bucket a_i^1 and their requirement tokens $a_{req,i}(t)$ in phase j_{th}(j=1,2,3…) and the sum of weights W^j of token buckets whose requirement are not met.

$$a_{able}^j(t) = \sum\left(a_i^j - a_{req,i}(t)\right),\ \ a_i^j \geq a_{req,i}(t) \tag{13}$$

$$W^j = \sum S_i^j, S_i^j \in \{S_i | a_{req,i}(t) > a_i^j\} \tag{14}$$

If $a_{able}^j(t) > 0$ and $W^j > 0$, there are remaining tokens can be obtained after the end of phase j_{th} allocation. So it will turn to the phase $j+1_{th}$ to allocate to the remaining tokens to unsatisfied token buckets, and execute (v.). Otherwise, it means that there are no remaining tokens or unsatisfied token buckets after the allocation j_{th}. Then it will turn to (vii.).

v. The $j+1_{th}$ phase allocation : redistribute the remaining tokens in phase j_{th} to the token bucket which tokens are not satisfied .The number of the tokens allocated in phase $j+1_{th}$ is as follows

$$a_i^{j+1} = \begin{cases} a_{req,i}(t), & a_i^j \geq a_{req,i}(t) \\ a_i^j + a_{able}^j(t)\frac{S_i^j}{W^j}, & a_i^j < a_{req,i}(t) \end{cases} \tag{15}$$

In the equation(15), when $a_i^j \geq a_{req,i}(t)$, it indicates that the token bucket has been allocated too many tokens in phase j_{th}, so it needs to be adjusted to its maximum demand tokens. When $a_i^j < a_{req,i}(t)$, it indicates that the token bucket is still in a token lacking status, and more tokens are still needed. And the tokens will get is $a_{able}^j(t)\frac{S_i^j}{W^j}$, it means that the tokens not only allocated by the remaining tokens of phase j_{th} $a_{able}^j(t)$ but also by the weight S_i^j of the token bucket and the sum of the weights W^j of all token buckets whose tokens is not satisfied.

vi. Turn to the step (iii), and continue to execute the allocation.

vii. Ending the allocation algorithm, and the number of allocated tokens $a_i(t)$ is the actual number of allocated tokens a_i^j which is token bucket i_{th} get at time t.

$$a_i(t) = a_i^j \tag{16}$$

3.3.3. Complete the Token Allocation Schedule for the Current Time

After the time Δt, the system will execute the token allocation arrangement for the next period, and refresh the information of each token bucket, and then turn to step 1).

4. Experiment Analysis

In order to verify the effectiveness of the method proposed in the article, we will carry out a simulation. For the sake of convenient analyzing, three services in the TCSN system are established. They are train control service, video service and ordinary data service.

We simulate the case with the overflow token bucket and the case without overflow token bucket respectively, and analyze the packet loss rate of both.

Through the simulation, we get the packet loss rate of the three services in two cases, so we can draw comparison charts as follows.

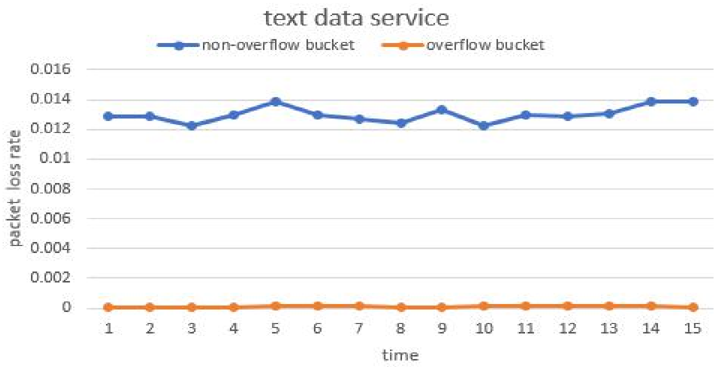

Figure 4: Comparison of packet loss rate for ordinary text data service.

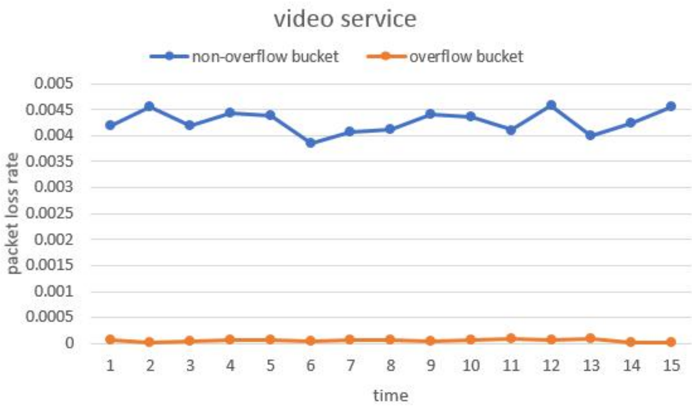

Figure 5: Comparison of packet loss rate for video service.

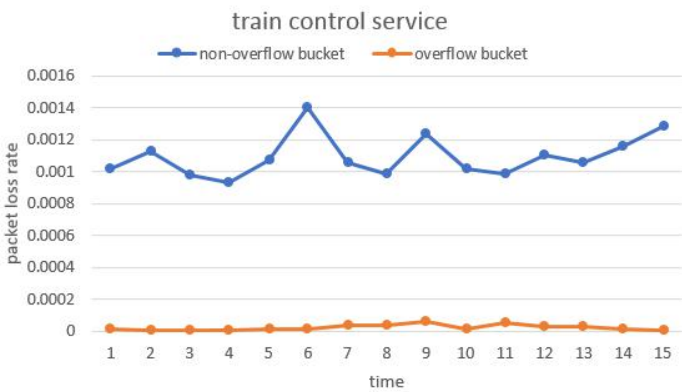

Figure 6: Comparison of packet loss rate for train control service.

It can be seen from the Figure 4, Figure 5, and Figure 6 that when there is overflow bucket, the packet loss rate of the three services is greatly reduced.

For analyzing more accurately, we calculate the average of the packet loss rate of each service separately, and the result as shown in Table 1.

Table 1: The average packet loss rate of different services in different case.

Part / Name	Ordinary text data	Video service	Train control
Non-overflow bucket	0.013019	0.001096	0.004269
Overflow bucket	0.0000998	0.0000228	0.0000559

As can be seen from the above table, the packet loss rate of the three services has been greatly reduced compared to the previous ones. So this algorithm can not only ensure the integrity of information transmission and avoid the waste of the tokens, but also improve the safety of the train and the passenger's internet experience.

5. Conclusions

In this paper, according to the situation of the TCSN system, and combined with the research status of the token bucket algorithm and the traffic model of the computer network, a smart token allocation strategy is proposed, which is a dynamic token allocation strategy based on weight. The core of the strategy lies in the weight of each token bucket. The weight not only reflects the priority of each token bucket but also integrates into the token allocation as a measurement factor. The entropy weight method is based on the three indicators to evaluate, which include the current number of tokens, the packet loss rate and the number of buffered packets. This strategy can redistribute the tokens of the overflow bucket so that the each packet can be quickly forwarded, and the packet loss rate is reduced. Therefore, it's a good reference for the design of TCSN.

Acknowledgments

This work was financially supported by National Science and Technology Support program (2015BAG14B01-05).

References

[1] Yangtao Li, "State and Development of TCN Train Network Technology"[J]. Microcontroller and Embedded System Application. Shanghai, pp.4-7, December 2012.

[2] Xue Bai, "Classification Identification and Modeling of Train Contol and Service Network"[D]. Southwest Jiaotong University. 2017.

[3] Xifeng Wang, "Research on Characteristic Analysis and Forest of Network Traffic"[D]. Northwest University. 2007.

[4] P. Orenstein, H. Kim and C. L. Lau, "Bandwidth allocation for self-similar traffic consisting of multiple traffic classes with distinct characteristics". IEEE Global Telecommunications Conference, San Antonio, TX, 2001, pp. 2576-2580 vol.4.

[5] Chunfeng Song, "Research on Self-Similarity of Network Traffic"[D]. Jilin Universoty. 2015.

[6] Hui Wang. "Research on The Fluid Flow Model for Self-Similar Networks Traffic and AQM Algorithm"[D]. Harbin Institute of Technology. 2013.

[7] N. Likhanov, B. Tsybakov and N. D. Georganas, "Analysis of an ATM buffer with self-similar ("fractal") input traffic," Proceedings of INFOCOM'95, Boston, MA, USA, 1995, pp. 985-992 vol.3.
[8] Zhengyuan Jia, Zhou Fan, Miaomiao Jiang, "Distribution Network Planning Based on Entropy Fuzzy Comprehensive Method", International Conference on Advanced Mechanical Engineering, Luoyang, 2010, pp.780-784.
[9] Lei Fu. "Study on the Assessment System of the Tourist Highway Based on the Entropy Weight Method"[D]. Chongqing Jiaotong University. 2014.
[10] Huixia Zheng. "Research on Work Safety Standardization Evaluation of Tobacco Distribution Center Based on Entropy Method"[D]. Xhengzhou University. 2012.
[11] Wenhong Liu, Xiaoliang Yang, Hongke Zhang, "Weighted and Fast Max-Min Fair Allocation Algorithm", Journal of Beijing Jiaotong University, Beijing, 2006, pp.33-35+47.
[12] Tassiulas L, Sarkar S. Maxmin fair scheduling in wireless and hoc networks[J]. IEEE Journal of select Areas Commun, pp.163-173, 2001.