MQSD: A Scheduling Mechanism of Mixed Queue in the Integrated Intelligent Network

Jiangshan Zheng1,2, Li Yang1,2*, Chengsheng Pan3 and Huaiyong Shi
1 School of Information Engineering, Dalian University, Dalian 116622, China
2 Communication and Network Laboratory, Dalian University, Dalian 116622, China
3 School of Electronics and Information Engineering, Nanjing University of Information Science & Technology, Nanjing 211800, China
*Corresponding author’s e-mail: yangli@dlu.edu.cn

Abstract. In the integrative intelligent network, the burst and heterogeneous characteristics of the aggregation node traffic lead to increased queuing delay and frame loss rate of the business flow. It is difficult to meet the quality of service (QoS) requirements of the differentiated business flow. In response to the above problems, a mixed queue scheduling based on demand service (MQSD) is proposed for the integrated intelligent network of heaven and earth. Different scheduling algorithms are used to schedule different business flows, which distinguishes the different needs of the business. The weight of business flow is calculated using the construction judgment matrix method, and is determined to be acceptable through a consistency test. By comparing the available service volume with the demanded service volume, the available service volume to the business flow will be dynamically adjusted. Establish a simulation experiment platform to schedule different types of randomly generated data streams. Simulation results show that the MQSD mechanism reduces queuing delay and frame loss rate caused by burstiness while meeting QoS requirements.

1. Introduction
The integrative intelligent network is based on the ground network and extended by the space network, covering natural spaces such as space, air, land and ocean, and providing information guarantee for the activities of space-based, air-based, land-based and sea-based users Infrastructure [1].

With the substantial increase in the computing capabilities, storage capabilities, and service transmission capabilities of nodes in the network, the world-ground integrated network continues to evolve toward the world-ground integrated intelligent network. After the business in the node is calculated, the invalid business flow is filtered, and the original large-capacity business flow is processed and turned into a small-capacity business flow. The content of the business is compressed, resulting in variable speed and burst characteristics of traffic, queuing delay and frame loss rate increase; the storage capacity of the node allows users to directly obtain request information from the cache node, speeding up the flow of traffic distribution, resulting in heterogeneous and mixed traffic characteristics, and different requirements for services such as queuing delay and transmission bandwidth.

Therefore, it is urgent to build a suitable heterogeneous link aggregation control method for traffic scheduling in the integrative intelligent network, combining the burst and heterogeneous characteristics of traffic, so as to meet the quality of services (QoS) for different service transmission requirements to reduce service queuing delay and frame loss rate.
2. Related work
Queue scheduling means that the sink node selects the next forwarded packet from the queue according to a certain rule, so that all input streams can share the output link bandwidth in a predetermined manner, occupy corresponding network resources, and meet their own required quality of service.

At present, there are the following algorithms that consider the burstiness of the service flow among the related queue scheduling algorithms. Literature [2] adds a token bucket control to each queue based on the weighted fair queuing algorithm (WFQ) [3], thereby alleviating the impact of burst traffic on delay and bandwidth. Literature [4] designed a prediction DWRR (P-DWRR) based on the self-similar characteristics of network traffic. In order to reduce the queuing delay and packet loss rate caused by the burst of network traffic, based the Deficit Weighted Round Robin (DWRR) [5], a self-similarity traffic is classified and predicted, and the queue weight and service quantum are dynamically adjusted to dynamically adjust the scheduling sequence. Literature [6] proposed a distributed serial scheduling structure that combines token buckets with a queue equalization algorithm based on proportional fairness.

In addition, there are several algorithms for considering QoS service classification. Literature [7] proposed a mixed queue scheduling on AFDX (MQSA), which classifies business flows into safety-critical business flows and non-safety-critical business flows. The first come first service algorithm (FCFS) [8], deficit round robin algorithm (DRR) [9] and strict priority algorithm (SP) [10] in parallel to ensure its real-time performance and fairness. Literature [11] proposed a differentiated service-based queue scheduling (P-VDWRR) algorithm, which combines the priority scheduling algorithm with the variable deficit weighted round-robin scheduling algorithm (VDWRR) [12], to provides different services for different types business data in the network. Literature [13] proposed a new prediction-based dynamic scheduling mechanism (PSS), which divides traffic into different categories and prioritizes them according to different QoS. Adjusted queue scheduling weight and buffer allocation dynamically to ensure delay and bandwidth performance. Some of these algorithms only consider the burstiness of the service flow, and some only consider the classification of the service flow. There is no algorithm that combines the two characteristics, it is difficult to ensure the overall utilization of network resources.

Based on the above analysis, this paper designs a mixed queue scheduling based on demand service (MQSD), which uses different scheduling algorithms for different business needs; at the same time, on the basis of the SP algorithm, the business flow weights are calculated by the construction judgment matrix method, and the dispatch volume model is established to dynamically adjust the service volume. Finally, a simulation verification system was built, and the performance simulation verification of the MQSD scheduling mechanism was completed by simulating the performance of queuing delay, bandwidth occupation and frame loss rate.

3. Queue scheduling based on the integrative intelligent network
In the integrative intelligent network, satellites have computing and storage capabilities, can carry payloads, and achieve effective service transmission. When designing an intelligent network queue scheduling algorithm based on this, it is necessary to consider not only the service flow from other nodes, but also the service flow stored by the aggregation node itself. The logic diagram is shown in Figure 1.
In Figure 1, $N : N = \{S_\alpha, D_\beta, I\}$ represents the satellite node in the intelligent network, $S_\alpha$ is the source node, $D_\beta$ is the destination node, and $I$ is the sink node. The traffic is $F : F = \{F_{S_\alpha}(t), F_{D_\beta}(t), F_I(t)\}$ in each node of the intelligent network at time $t$, where $\alpha, \beta \in [0, n]$.

Assume that queue scheduling occurs at time $t$, and the service flow of each node is not transmitted at time $t-1$. Since the integrated intelligent network node has storage capacity, the traffic in each node is $F_{S_\alpha}(t-1)$, $F_{D_\beta}(t-1)$, $F_I(t-1)$ at time $t-1$; the traffic in aggregation node $I$ is $F_I(t) = F_I(t-1) + F_{S_\alpha}(t-1)$ at time $t$. When $[t-1, t]$, the service flow stored by the sink node and the service flow of the source node are classified into delay-sensitive, bandwidth-sensitive, and other types of service flows according to different QoS requirements through the classifier. Then they are scheduled by the scheduler, and the resulting service flows are forwarded to different destination nodes. The flow relationship is shown in formula (1).

$$F_I(t) = F_{S_\alpha}(t) + F_{D_\beta}(t) + F_I(t)$$

$$= F_I(t-1) + F_{S_\alpha}(t-1)$$

4. MQSD scheduling mechanism

The basic idea of the MQSD scheduling mechanism proposed in this paper is as follows. In view of the heterogeneous and mixed characteristics of the integrated intelligent network traffic, different types of service flows have different requirements for QoS, so different scheduling algorithms are selected. According to the burst characteristics of intelligent network traffic, three parameters are selected: the priority of the service flow, the burst degree and the maximum frame length allowed to be sent. Use the construction judgment matrix method to set the weights of these parameters, and conduct a consistency test to determine that the weights are acceptable. At the same time, by comparing the available service...
volume with the demanded service volume, dynamically adjust the service flow to obtain the service volume. The logic diagram of the queue scheduling mechanism is shown in Figure 2.

![Figure 2 Logical diagram of MQSD scheduling mechanism](image)

4.1. Parameter selection

**Definition 1** Suppose $q_{size_i}$ is the length of queue $i$ at the current moment, $q_{max_i}$ is the maximum length of queue $i$, and $q_{avg_i}$ is the average length of queue $i$, then the burstiness of the service flow meets

$$d_{burst_i} = \frac{q_{size_i} - q_{avg_i}}{q_{max_i}}$$

(2)

The average queue length is calculated by an exponentially weighted moving average low-pass filter, which reflects the network congestion to a certain extent, as shown in formula (3):

$$q_{avg_i} = (1 - f_i)q_{avg_i} + f_i \cdot q_{size_i}$$

(3)

In the formula: $f_i$ is the low-pass filter coefficient. When the value is 0.01, the instantaneous jitter of the queue can be effectively reduced [14], thereby obtaining lower delay jitter.

Assuming $0 < i < n$, $p_i$ is the priority of the service flow in queue $i$. The larger the $p_i$ value, the higher the priority of the service flow. $d_{burst_i}$ is the burst degree of the service flow in queue $i$, and $l_{max_i}$ is the maximum frame length allowed to be sent in queue $i$.

Priority, burstiness and maximum frame length allowed to be sent are different attributes of the service flow, simple addition processing cannot comprehensively reflect the effect of each parameter. So normalize its corresponding parameters. The original data is mapped to the $[0, 1]$, so that the indicators are in the same order of magnitude. The normalization function is:

$$\bar{x} = x_i / \sum_{i=1}^{n} x_i$$

(4)

$x_i$ is one of the three parameters. The three parameters are service flow priority parameters, burstiness parameters, and the maximum frame length allowed to be sent. The parameter obtained after normalization is infinite, and the parameter is $\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_n \in [0, 1]$, where $n$ is the number of queues.

After normalization, calculate the weighted values of all parameters:
\[ c_i = \omega_1 \cdot \overline{p}_i + \omega_2 \cdot \overline{d}_{\text{bursty}} + \omega_3 \cdot \overline{I}_{\text{max}}. \]  \hspace{1cm} (5)

Where \( \overline{p}_i \), \( \overline{d}_{\text{bursty}} \), and \( \overline{I}_{\text{max}} \) are the priority parameters, burstiness parameters and the maximum frame length allowed to be sent in the normalized service flow in the \( i \) queue, \( \omega_1, \omega_2, \omega_3 \) is the weight of each parameter. By sorting the obtained weighted value \( c_i \), the corresponding service flows are scheduled in order from largest to smallest.

4.2. Dynamically adjust weights

The selection strategy of business flow weight adopts the construction judgment matrix method. First, compare different indicators in pairs, use important scale standards to measure, obtain relatively important scale parameters to form an order square matrix which dimension is \( n \). Then calculate the weight of each indicator by the characteristic root method, Finally verify the consistency of the matrix by checking the coefficient CR Perform inspection. The important scales between different elements are shown in Table 1.

| Important scale | Important scale meaning |
|-----------------|-------------------------|
| 1               | Two indicators are of the same importance |
| 3               | The former indicator is slightly more important than the latter |
| 5               | The former indicator is obviously more important than the latter |
| 7               | The former indicator is very important than the latter |
| 9               | The former indicator is more important than the latter |

According to the relative important scale parameters of each index, the matrix \( A \) is constructed as shown in formula (6).

\[
    A = \begin{bmatrix}
        a_{11} & a_{12} & \cdots & a_{1n} \\
        a_{21} & a_{22} & \cdots & a_{2n} \\
        \vdots & \vdots & \ddots & \vdots \\
        a_{n1} & a_{n2} & \cdots & a_{nn}
    \end{bmatrix}
\]  \hspace{1cm} (6)

Among them, \( a_{mn} \) is the relative importance of the \( m \) index to the \( n \) index, and \( a_{mm} > 0, a_{mm} = \frac{1}{a_{mm}}, a_{nn}, a_{nn} = 1 \). Then

\[
    (nE - A)\omega = 0
\]  \hspace{1cm} (7)

Where \( E \) is the identity matrix. According to the characteristic root method:

\[
    \lambda_{\text{max}} \omega = \Lambda \omega
\]  \hspace{1cm} (8)

Among them, \( \lambda_{\text{max}} \) is the maximum eigenvalue of matrix \( A \). The corresponding feature vector is \( \omega = [\omega_1, \omega_2, \ldots, \omega_n] \), which is the weight vector.

In order to ensure the logic of the relative important scales of different indicators, the test coefficient \( CR \) \(^{[15]} \) will be introduced to test the consistency of matrix \( A \). That is, matrix \( A \) determines the allowable range of inconsistency. The test coefficient \( CR \) is the ratio of the consistency index \( CI \) and the random consistency index \( RI \), namely

\[
    CR = \frac{CI}{RI}
\]  \hspace{1cm} (9)

In the formula, \( CI = \frac{\lambda - n}{n - 1} \). When \( CI = 0 \), there is complete consistency. When \( CI \)'s value is close to 0, there is satisfactory consistency. The larger the \( CI \)'s value, the more serious the inconsistency, and
the greater the judgment error. The random consistency index is
\[ RI = \frac{CI + CI_2 + \cdots + CI_n}{n}. \]
The RI’s value is affected by the order of the judgment matrix. In general, the larger the order of the matrix, the greater the probability of random deviation from consistency. The corresponding RI’s values of the matrix order are shown in Table 2. For the test coefficient CR, if \( CR < 0.1 \), the matrix A is considered to pass the consistency test. Otherwise, the relatively important scale should be reconfirmed.

### Table 2 Matrix order corresponds to RI value

| Matrix order | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 |
|--------------|----|----|----|----|----|----|----|----|----|----|
| RI           | 0  | 0  | 0.58 | 0.90 | 1.12 | 1.24 | 1.32 | 1.41 | 1.45 | 1.49 |

#### 4.3. Dynamically adjust service volume

When scheduling the frames in queue \( i \), in addition to the scheduling order, the amount of scheduling should also be considered, that is, how much to schedule each time.

Assuming \( 0 < i < n \), in queue \( i \), \( Q_i \) is the current service flow obtained service volume, \( Q_{ir} \) is the available service volume, \( Q_{ri} \) is the required service volume of the service flow, and \( Q_{it} \) is the degraded service volume of the service flow.

This article will compare the service volume that can be provided with the service volume required to determine the current business flow to obtain the service volume \( Q \). When the \( Q_{it} \) is consumed, the remaining part of the current service flow will be degraded. That is, this part of flow enter the queue with the lowest priority and wait for the next round of scheduling. \( Q_{it} \) will be dynamically adjusted, its quantitative index is the time slice \( T \), and the \( Q_{it} \) quantitative index is the execution time \( t_i \), that is:

\[ Q_{it} : Q_{it} - a \cdot T, \quad Q_{ir} : Q_{ir} \sim b \cdot t_i. \]

There is a number set \( X = \{X_i : i = 1, 2, 3, \ldots, n\} \). All data in the logarithm set are sorted in ascending order as follows:

\[ X' = \{X'_i : i = 1, 2, 3, \ldots, n\} \quad (10) \]

The median is the number in the middle of the set of numbers arranged in order. It is not affected by the two extreme values of the largest and the smallest. Changes in some data have no effect on the median.

For the number set \( X \), the formula for solving the median \( X_M \) is as follows:

\[ X_M = \begin{cases} X'_{(i+1)/2}, & n \text{ is odd number} \\ \frac{X'_{(i+1)/2} + X'_{(i+2)/2}}{2}, & n \text{ is even number} \end{cases} \quad (11) \]

For the selection of time slices, a dynamic adjustment strategy is adopted. In view of the fact that the description of the median log set is more representative, the calculation method of the time slice \( T \) decided to use the method of calculating the median execution time of all queues. The calculation formula is as follows:

\[ T = M = \begin{cases} \frac{t_{i+1}}{2}, & n \text{ is odd number} \\ \frac{t_{i/2} + t_{i/2+1}}{2}, & n \text{ is even number} \end{cases} \quad (12) \]

In the formula: \( M \) is the median execution time of service flows participating in scheduling; \( MM \) is the kth execution time of all service flows in ascending order; \( n \) is the current number of service flows participating in scheduling.

While analyzing the execution time \( t_i \), the system assigns the default time slice \( T_i \) to the queue. Therefore, the calculation formula for time slice \( T \) will be updated as follows:
The amount of services that can be provided is shown in formula (14):

\[ Q_T = \begin{cases} Q_\alpha - a M, & M \geq T_0 \\ Q_\beta - a T_0, & M < T_0 \end{cases} \]

(14)

For formulas (13) and (14), it can be seen that the time slice \( T \) has been dynamically adjusted, that is, its value is the maximum value of the median \( M \) and the default time slice \( T_0 \). So \( Q_T \) is also dynamically adjusted. The dynamic adjustment of \( Q_T \) maximizes the available service volume of the current service flow, which provides a guarantee for delay-sensitive service flows and bandwidth-sensitive service flows to ensure their respective QoS.

At this time, the service volume obtained by the queue is as shown in formula (15).

\[ Q(t) = \begin{cases} Q_\alpha(t) - b t_i, & t_i \leq T \\ Q_\beta(t) - a T, & t_i > T \end{cases} \]

(15)

4.4. Scheduling process

The MQSD scheduling mechanism distinguishes the different needs of the business, and dynamically adjusts the weight and scheduling volume. The steps are as follows:

Step1: When the frame arrives at the sink node, the classifier determines the type of the current frame. If it is a delay-sensitive type, go to Step2, if it is a bandwidth-sensitive type, go to Step3, otherwise, go to Step4;

Step2: All frames are virtualized into a virtual output queue \( VOQ_i \), of which \( i = 1 \). The first come first service scheduler fetches the frame from \( VOC_{31} \) and forwards it. The frame is buffered in queue 3 with a priority of 4, go to Step5;

Step3: All frames are virtualized into three virtual output queues \( VOQ_{2i} \), of which \( i = 1, 2, 3 \). The weighted round-robin scheduler fetches the frame from \( VOC_{2i} \) and forwards it. The frame is buffered in queue 2, with priority 3, go to Step5;

Step4: All frames are virtualized into three virtual output queues \( VOQ_{1i} \), of which \( i = 1, 2, 3 \). The balance round-robin scheduler fetches the frame from \( VOC_{1i} \) and forwards it. The frame is buffered in queue 1, the priority is 2, go to Step5;

Step5: Analyse the parameters of the frames in each queue: priority \( p_i \), burst degree \( d_{burst} \), allowable maximum frame length \( l_{max} \), execution time \( t_i \), default time slice \( T_0 \). Calculate the median \( M \) according to the execution time during service flow scheduling;

Step6: Compare the size of \( T_0 \) and \( M \). If \( T_0 \) is less than \( M \), then time slice \( T \) is equal to \( M \), otherwise, time slice \( T \) is equal to \( T_0 \).
Step7: Compare the size of $t_i$ and $T$. If $t_i$ is greater than $T$, go to Step8, otherwise go to Step9;
Step8: The current queue allocates service volume $Q_i$ to the current business flow, and go to Step9;
The remaining service amount $Q_L$ is allocated to the downgrade queue and go to Step5;
Step9: Normalize the parameters: $p_i$, $d_{burst}$, and $l_{max}$, respectively, and obtain the weight $c_i$ by constructing the judgment matrix;
Step10: Sort $c_i$ in descending order and output frames in sequence. Determine whether all queues are empty. If yes, the scheduling is complete; otherwise, go to step 1.

5. Simulation results and performance analysis

5.1. Simulation platform and parameter setting
The world-ground integrated intelligent network has a large number of nodes and complex node types. The queue scheduling mechanism studied in this paper is the scheduling of business flows in a node, so in order to simplify the model, the network topology used in this simulation is shown in Figure 3. Simulation verification is carried out using NS2 simulation software.

A1, A2, and A3 are three satellite source nodes respectively. The service streams transmitted to node B are delay-sensitive service streams, bandwidth-sensitive service streams, and other service streams. The sending time interval and sending rate are shown in Table 3. B is an aggregation node with a service rate of 60Mbps, which has a classification function. Because the integrated intelligent network node of the world has a storage function and is a simulation storage function, when it is simulated in the NS2 software, the aggregation node B will also generate a service flow. There is no type restriction. After classification, it participates in scheduling together with the service flow generated by the source node. C is the destination node.
Table 3 Source node parameter settings

| Source node | A1   | A2   | A3   |
|-------------|------|------|------|
| Sending interval (ms) | 15   | 10   | 10   |
| Send rate (Mbps) | 10   | 20 (variable) | 7    |

When the source nodes of A1, A2, and A3 send service flows to the sink node B, they are buffered in queues 3, 2, and 1. The priority of queues 3, 2, and 1 are 4, 3, and 2, respectively, and the priority of the degraded queue is 1. The importance of various parameters in different types of business flows is compared in pairs, and the weights of various indicators are obtained according to the obtained important metrics and judgment matrix. The priority of each service stream, the burst degree and the weight setting of the maximum frame length allowed to be sent are shown in Table 4.

Table 4 Simulation attribute weight setting

| Data flow   | Weight vector      |
|-------------|---------------------|
| Delay sensitive | (0.56, 0.24, 0.20)T |
| Bandwidth sensitive | (0.23, 0.56, 0.21)T |
| other kind   | (0.33, 0.33, 0.34)T |

According to the attribute weight setting, combined with formula (8) to calculate: CRd=0.05, CRb=0.07, CRo=0.08. Since the three types of business flow test coefficients are all less than 0.1, their weight vector estimates are all within the error range.

5.2. Simulation results and analysis

In order to reflect the advantages of the queue scheduling mechanism in this paper, the strategy proposed in this paper is compared with the dynamic packet scheduling mechanism based on prediction (PSS(13)) and the hybrid queue scheduling mechanism (P-VDWRR(11)). The comparison result will be reflected in the three performance aspects of queuing delay, bandwidth occupation, and frame loss rate.

5.2.1. Time delay performance simulation and analysis

This article will take the delay-sensitive service flow as the analysis sample to analyze the queuing delay performance of this mechanism, PSS mechanism and P-VDWRR mechanism. The average delay simulation results of the three scheduling mechanisms for the service flow are shown in Figure 4.
For delay-sensitive service flows, the scheduling mechanism in this paper is 5.84% lower than the PSS mechanism and 2.9% lower than the P-VDWRR mechanism. This is mainly because the PSS mechanism dynamically adjusts the queue scheduling weight by real-time counting the number of active flows in each class. In the P-VDWRR mechanism, the token bucket mechanism limits the service flow, and neither considers the suddenness of the service flow characteristics. However, when the service flow is burst, the buffered frames in the queue will increase sharply, resulting in an increase in the delay of the frames in the queue. In the mechanism of this article, when dynamically setting the weight and service volume, the burstiness is comprehensively considered, and the service flow is scheduled before the burst, thereby reducing the delay. At the same time, for the delay-sensitive service flow, the FCFS scheduling algorithm is used for delay-sensitive business flows, which has good delay performance.

5.2.2. Bandwidth performance simulation and analysis
Bandwidth refers to the amount of data that can be transmitted per unit time. In this paper, the sending rate of bandwidth-sensitive service streams is gradually increased from 20MB/s to 70MB/s. For different scheduling mechanisms, the simulation results of bandwidth occupancy curves are shown in Figure 5.
As can be seen from the figure, compared to the PSS scheduling mechanism and the P-VDWRR scheduling mechanism, the number of frames is constantly changing with the increase of the service flow sending rate. The scheduling mechanism in this article can make better use of bandwidth, and its bandwidth utilization is better than that of PSS. The scheduling mechanism has increased by 6.99% on average, which is an average increase of 10.58% compared to the P-VDWRR scheduling mechanism. This is mainly because P-VDWRR only retains the advantages of the WRR scheduling mechanism, and its bandwidth performance is similar to the WRR scheduling mechanism. The PSS mechanism can only dynamically adjust the queue buffer size. In the scheduling mechanism of this article, after bandwidth-sensitive service flows are scheduled through WRR, a downgrade strategy is introduced, so that some frames with demanded service volume exceeding the available service volume can be scheduled again, that is to say, its weight is increased and assigned to more bandwidth, more services. The scheduling mechanism in this article has obvious advantages in bandwidth performance.

5.2.3. Frame loss rate performance simulation and analysis
The frame loss rate refers to the ratio of lost frames to the total number of frames sent during data transmission. When the network is congested, frame loss is an inevitable phenomenon. Therefore, when the buffer length is fixed, the frame loss rate should be reduced as much as possible. For different scheduling mechanisms, the frame loss rate simulation curve is shown in Figure 6.
It can be seen that the frame loss rate of this mechanism is 7.9% lower than that of the PSS mechanism, and 14.2% lower than that of the P-VDWRR mechanism. This is mainly because the burst of traffic will cause more frames to be buffered in the queue. When the number of frames is greater than the buffer length, some frames will overflow, that is, frame loss. The mechanism in this paper introduces the service flow burstiness parameter when dynamically setting the scheduling weight, and performs scheduling before its burst, so it will reduce the probability of frame loss caused by data burst. However, the PSS mechanism and the P-VDWRR mechanism do not consider the burst characteristics of traffic.

6. Conclusion
Based on the integrative intelligent network architecture, in order to solve the problem of multiple changes in heterogeneous traffic and sudden changes in the characteristics of large-span traffic in the integrative intelligent network, the MQSD scheduling mechanism is proposed, and different scheduling algorithms are used to distinguish different business needs. On this basis, the construction judgment matrix method is used to dynamically adjust the business flow weights, the dispatch volume model is constructed to dynamically adjust the service volume, and simulation verification is performed. The simulation results show that the algorithm in this paper guarantees the QoS of different types of service flows and reduces the burstiness of service flows. However, when constructing the scheduling volume model in this paper, the selection strategy of the time slice of the quantitative index that can provide the service volume uses the median method. The median will not be affected by the maximum or minimum value in the number set. The appearance of the maximum or minimum value is fair to other cohorts and has no effect. However, for the queues with extremely or extremely low values, the delay increases and the burstiness is not improved due to the inability to complete the scheduling for a long time. Therefore, the follow-up work will focus on the time slice selection strategy to ensure the QoS of each business.

Acknowledgments
This work is supported by the National Natural Science Foundation of China under Grants 61722105, 61931004.
References

[1] SUN C H. Research Status and Problems for Space-based Transmission Network and Space-ground Integrated Information Network [J]. Radio Engineering, 2017, 047(001):1-6.

[2] MENG G D. Research on the detection queue scheduling algorithm of the video surveillance intelligent detection system[D]. Shandong: Ocean University of China, 2015.

[3] LI B Q, ZHANG S, WANG Z W. Performance analysis and improvement of WFQ and WRR scheduling algorithms [J]. Transaction of Beijing Institute of Technology, 2015, 35(03):316-320.

[4] WEI D B, SHEN T, YANG L. Network queue scheduling algorithm based on self-similar traffic level grading prediction. [J]. Journal on Communications, 2020, 41(04):182-189.

[5] Bök, P B, Kohls, Katharina. I-DWRR : an insolvency enabled scheduling scheme extending deficit weighted round robin[C]//Globecom Workshops. IEEE, 2012.

[6] Wan P , Zhan Y , Pan X. Scheduling algorithm for the multiple rovers' access to single orbiter on the Mars relay communication links[J]. International Journal of Satellite Communications and Networking, 2019.

[7] LIU J. Research on Optimization of AFDX Real-time Using Mixed Queue Scheduling Policy[J]. Computer Engineering, 2015, 041(011):135-141.

[8] Vankka J. Performance of Satellite Gateway over Geostationary Satellite Links[C]//Military Communications Conference, 2013: 289-292.

[9] ZHONG Y J, YU X S, ZHANG Q S, Implementation and optimization of congestion control and queuing scheduling in 10 Gigabit network [J]. Electronic Design Engineering, 2017, 25(24): 82-87.

[10] GONG S L, WANG C, FENG Y G. Substation LAN communication queue scheduling strategy based on improved weighted fair queue [J]. Automation of Electric Power Systems, 2015, 39(04):76-81.

[11] CUI Y, WU J P, XU K. A Survey of Internet Quality of Service Routing Algorithm Research [J]. Journal of Software, 2002, 13(11).

[12] ZHONG Y J, YU X S, ZHANG Q S. Implementation and optimization of congestion control and queuing scheduling in 10 Gigabit network [J]. Electronic Design Engineering, 2017, 25(24): 82-87.

[13] Sharma R , Kumar N, Gowda N B , et al. Packet Scheduling Scheme to Guarantee QoS in Internet of Things[C]//Springer US, 2018.

[14] LI B, CHEN J S, WANG Y. Demand response business scheduling strategy based on fuzzy logic improved weighted fair queue[J]. Automation of Electric Power Systems, 2020, 44(02):60-69.

[15] YANG L, SUN J, PAN C S. LEO multi-service routing algorithm based on multi-objective decision making.[J] Journal on Communications, 2016, 37(10):25-32.