# Concurrent control algorithm for large-scale remote multi-user information management system

Dawei Zhao, Network Center, Baicheng Vocational and Technical College, Baicheng, 137000 China

Received: June 28, 2021. Revised: December 14, 2021. Accepted: January 3, 2022. Published: January 5, 2022.

Abstract—When the current concurrency control algorithm is used to control of the multi-user information management system, the system's channel transmission capability is low, and the time it takes is long. In this paper, a concurrency control algorithm for large-scale remote multi-user information management system is proposed. According to the average use rate of the large-scale remote multi-user information management system, the concurrency control structure and state of the system are analyzed and judged; Through the analysis of the results, the delay of data link layer in multi-user information management system is carried out modeling; Combined with the queuing delay and accessing delay, the large-scale remote multi-user information management system control can be realized. Experimental results show that the channel utilization rate of the proposed algorithm is over 98.3%, which can transmit large amounts of information in a relatively short time and concurrency control of information management system. Therefore, the proposed algorithm has high channel utilization and efficiency of information transmission.

Keywords—Concurrency control, delay, information management system, remote.

#### I. INTRODUCTION

WITH the acceleration of information process, the computer and the Internet have been widely used [1]. In large-scale remote multi-user information management systems, multiple users need to access shared objects, such as collaborative documents, primitives of whiteboard in real-time data conferences, etc. [2], [3]. Therefore, the system must respond to the user's access operations in real time. In an information management system, the server site is responsible for storing objects shared by a collaborative user in an activity. User's operation on each object must be sent to the server. The execution server will copy the shared objects to the local users' terminals in multicast way, so as to maintain the consistent of

shared objects at each user end [4]. In large-scale remote multi-user information management systems, there may be concurrent access to an object by multiple users at the same time, and conflicts are unavoidable. The network transmission delay and unreliability between multiple machines, as well as the factors of multi-user collaboration, will add a new possibility to the conflict [5]. Therefore, it is necessary to analyze and study the concurrency control algorithms in large-scale remote multi-user information management system [6].

An adaptive concurrency control algorithm based on conflict rate prediction was proposed by Fan and Zhuang [7]. The algorithm was divided the concurrency control process into two stages: information collection and strategy selection. The information collection stage ensured serializable transaction execution by using a priori transaction queue and collected the transaction execution state of the system using the cyclic conflict queue. The strategy selection stage was to use the improved weighted moving average method to predict the next stage conflict rate on the cyclic conflict queue, and according to the bidirectional threshold, to make concurrency control of the information management system, to complete the concurrency strategy in next stage. When using this algorithm to control the multiuser information management system, the time of information transmission to the client is longer and the efficiency of information transmission is poor. Zeng and Li proposed a concurrency control algorithm for embedded information management system [8]. In the algorithm, the parameters of resource requirements and task urgency in the embedded information management system were taken as the input variable of the neural network. The priority of control data was taken as output variables, and it was determined by dynamic weight algorithm and the concurrency control model of data was built to achieve concurrency control of multi-user information management system. This algorithm is more complex and easy to error, resulting in the congestion of the channel in the multi-user information management system, and the channel utilization rate of the system is low. In addition, a multi-path concurrent transmission data control algorithm was proposed by Du et al. [9]. Based on the analysis of the influence factors on

the path performance evaluation model, the evaluation model of transmission path performance was given. The buffer feedback was used to dynamically adjust the transmission ratio of the path, reduce the influence of the path with poor transmission performance on the overall throughput, and complete the concurrency control of the multi-user management system. The steps of this method are more, the time of calculation is long, and the efficiency of information transmission is slow. Li et al. also proposed a parallel control algorithm based on multi-core [10]. The algorithm combined multi-core and multi-threading technology to improve the traditional serial algorithm, enabling the computation dependent process to parallelize, so as to complete the concurrency control of multi-user information management system. However, the algorithm cannot transmit information through multiple paths, and the utilization rate of the channel is low. To sum up, a concurrency control algorithm for large-scale remote multi-user information management system is proposed in this paper, which is mainly structured as follows:

DOI: 10.46300/9106.2022.16.7

(1) The concurrency control structure and state of the system are analyzed and judged according to the average utilization rate of the large-scale remote multi-user information management system.

(2) The queuing delay and access delay of the large-scale remote multi-user information management system are carried out modeling.

(3) The concurrency control of the system is completed according to the queuing delay and the access delay of the large-scale remote multi-user information management system.

(4) Experimental results and analysis. The overall effectiveness of the proposed algorithm is to verify from two aspects of channel utilization and information transmission efficiency, respectively.

# II. CONCURRENCY CONTROL STRUCTURE FOR LARGE-SCALE **REMOTE MULTI-USER INFORMATION MANAGEMENT SYSTEM**

According to the average utilization ratio of large-scale remote multi-user information management system, the concurrency control structure and state of the system are analyzed and judged, to provide data information for queuing delay and input delay of computing system [11], [12]. The generation and arrival of data frames in the system are subject to Poisson distribution. The arrival time interval and processing time of the data frames are subject to the negative exponential distribution. The transmission processing of the data frame for the communication server conforms to the parallel standard service model in the queuing theory [13].

For the communication server in the system, the ratio between the average arrival number  $\varepsilon$  of the data frame and the average number of processed data  $\eta$  in the same time interval is called the data processing intensity B, that is, the average utilization rate of the system. B is the ratio of the data processing intensity B to the gateway number.

$$B_s = \frac{\varepsilon}{\eta} \tag{1}$$

The smaller the value of B is, the higher the efficiency of the communication server is, and vice versa. When B > 1, the communication server is very poor and is in a blocking state. Therefore, when designing the concurrency processing algorithm of the communication server, it is necessary to ensure that the data processing strength must be less than 1, and the infinite queue will not appear. The structure of the concurrency control algorithm in the system is shown in Fig. 1.



Fig. 1 Structure of concurrency control algorithm

The concurrency control structure is divided into three parts, namely peripherals, communication servers and clients: The peripherals are composed of gateways, which are used to interconnect two high-level networks with different protocols. At the same time, they provide filtering and security functions; the main function of the communication server is to save Threads, real-time database receiving, sending, and parsing threads; What the customer reflects in this structure is the demand. This structure can effectively ensure the consistency of the data, correctly coordinate multiple clients' access to the data at the same time, solve the problem of resource competition, and ensure the integrity of the data.

# III. MODELING OF DATA LINK LAYER DELAY OF MULTI-USER INFORMATION MANAGEMENT SYSTEM

The delay of data link layer in large-scale remote multi-user information management system is mainly composed of two parts: queuing delay and accessing delay [14]. Delay does not only mean that data does not arrive on time, but also reflects the overall performance of large-scale remote multi-user information management system, such as channel congestion, etc. [15].

# A. Queuing Delay

Queuing delay is defined as the time that the data frames go from the cache queue to leave the cache queue. When the system is processed at a certain speed, the queuing delay of the data frame depends mainly on the number of queued frames before the frame. New data frames into the cache queue of data link layer may be faced with one of the following two situations:

(1) The cache queue is empty, and the data frame can be taken the first service.

(2) Other data frames in the cache queue are queuing, and the data frame needs to wait for other data frames to be sent successfully to receive service.

Whether the cache queue is empty depends on the amount of data and the ability of the system to process data. In a certain period of time, if the processing rate of the system is greater than

the rate of the data frames entering the cache queue, the cache queue is empty. Otherwise, the cache queue is not empty. Therefore, the probability that the cache queue not empty can be expressed as:

$$\rho = \frac{\lambda}{\mu} \tag{2}$$

where,  $\lambda$  is the average rate of arrival of data frames,  $\mu$  is the average departure rate of data frames, and  $\rho$  is the probability that the cache queue is not empty.

According to the Little theorem, when the cache queue is not empty, the formula for calculating the average number  $\overline{L}$  of the data frames is:

 $\overline{L} = \lambda t \tag{3}$ 

In the formula, t represents the average time required to send a data frame, and it can be understood as the time interval between the two transmissions of the data frames.

When the cache queue is empty, the newly arrived data frame can be taken the first service, but it may also need to wait for time *t*. The reason is that, although the cache queue is empty, the system may have just taken the only data frame from the queue and are serving it. Therefore, when the queue is empty, the average time to wait for the newly arrived data frames is (0+t)/2.

The general queuing delay calculation formula for large-scale remote multi-user information management system is as follows:

$$T = \frac{(1-\rho)t}{2} + \rho \bar{L}t \tag{4}$$

In the formula,  $1-\rho$  indicates the probability that the cache queue is empty; t/2 indicates the average waiting time of the data frames when the queue is empty;  $\rho$  indicates the probability of the cache queue is not empty, and  $\overline{L}t$  is the average time of the data frames when the queue is not empty.

According to the (2) and (4), the queuing delay of the data link layer in the system can be obtained as follows:

$$T = \frac{\lambda^2 t^2}{\mu} + \frac{(\mu - \lambda)t}{2\mu}$$
(5)

The probability that the cache queue is not empty is calculated on the basis of the processing rate of the system and the rate of data frames entering the cache queue. When the cache queue is not empty, the average number of data frames is calculated according to the Little theorem, and the total queuing delay of the information management system is obtained, and then the queuing delay of data link layer is calculated.

#### B. Accessing Delay

The accessing delay depends on the channel usage mechanism of large-scale remote multi-user information management system. The principle of data frame transmission in large-scale remote multi-user information management system is illustrated in Fig. 2. The site needs to send competitive channel usage before sending the data. Therefore, the accessing delay is defined as the time consumed by the use of the site's



Fig. 2 Process of packet transmission

In Fig. 2, *ACK* is a confirmation frame, *DIFS* is a distributed inter-frame interval, *EIFS* is an extended inter-frame interval, *SIFS* is a short inter-frame interval, and  $\delta$  is the transmission delay. If the data frame is successful after *i* times of *RTS*/*CTS* attempts, the access delay *D* can be expressed as:

$$D = \sum_{j=1}^{i} D_j + (i-1) \cdot T_{timeout}$$
(6)

In the formula,  $T_{timeout}$  indicates the time allowed to send corresponding timeouts, and  $D_j$  indicates the waiting time for *j* th backoff, that is, the time from the start counter to the time when the counter is reduced to 0, and *i* is constant.

The formula for calculating the conditional expectation value E(D) of D is:

$$E(D \mid D_1, D_2, \dots, D_{R_{\max}}) = \sum_{i=1}^{R_{\max}} P_i \cdot \left[ \sum_{j=1}^{i} D_j + (i-1) \cdot T_{timeout} \right]$$
(7)

where,  $R_{\text{max}}$  is the upper limit of the retransmission number, and  $P_i$  indicates the probability of a data frame sending success after *i* times of attempts.

Set p be the probability of a collision for a data frame, and  $P_i$  can be expressed as:

$$P_i = p^{i-1} \cdot (1-p) \tag{8}$$

According to the full expectation formula, the mean value of D is expressed as:

$$E(D) = E(E(D | D_1, D_2, \dots, D_{R_{\max}}))$$
  
=  $D_{\alpha} + \sum_{i=1}^{R_{\max}} (p^{i-1} - p^{R_{\max}}) \cdot E(D_i)$  (9)

In the formula,  $D_{\alpha}$  is the full expected parameter, and the calculation formula of  $D_{\alpha}$  is as follows:

$$D_{\alpha} = \frac{p}{1-p} \cdot (1-p^{R_{\max}} - R_{\max} p^{R_{\max}-1} + R_{\max} \cdot p^{R_{\max}}) \cdot T_{timeout}$$
(10)

According to the process shown in Fig. 2,  $D_i$  can be expressed as:

$$D_i = \sum_{j=1}^{W_i} d_j + W_i \cdot t_{slot}$$
<sup>(11)</sup>

where,  $d_j$  indicates the random freezing time at the *j* th update points when the site updates the backoff timer value. When the channel states are free, collision and successful transmission, the values of  $d_j$  are 0,  $T_c$ , and  $T_s$  respectively.  $T_s$  is the time that the site successfully booked the channel and sent the data. (13)

 $T_c$  is the conflict time in the site booking channel;  $t_{slot}$  represents the length of the backoff time slot; And  $W_i$  is the initial value of the backoff timer for the i th retransmission.

 $T_c$ ,  $T_s$  and  $W_i$  can be expressed as:

 $W_i = uniform(0, 2^i w - 1)$ 

In the formula, *EIFS* is the extended inter-frame interval, *RTS* represents the request sending,  $\delta$  is the propagation delay, *DIFS* is the distributed inter-frame interval, *SIFS* is the short inter-frame interval, *CTS* represents the allowed transmission, *ACK* is the confirmation frame, and *w* is the smallest competition window.

 $E(D_i)$  in (9) can be derived from (14) and (17).

$$E(D_{i}) = E\sum_{j=1}^{W_{i}} d_{j} + E(W_{i} \cdot t_{slot})$$
(14)

$$E\sum_{j=1}^{W_{i}} d_{j} = E(W_{i}) \cdot E(d_{j})$$
(15)

$$E(d_j) = T_c \cdot P_c + T_s \cdot P_s \tag{16}$$

where,  $P_s$  is the probability of the successful transmission of the data frame, and  $P_c$  is the probability of the collision of the data frame, which can be calculated by (17).

$$\begin{cases} P_s = n\tau (1-\tau)^{n-1} \\ P_c = 1 - (1-\tau)^n - n\tau (1-\tau)^{n-1} \end{cases}$$
(17)

where, the number of sites in the user information management system is n, and the probability of data transmission is  $\tau$  for the time slot selected by a site. The value of  $\tau$  can be obtained by (18) and (19).

$$\begin{cases} \tau = \frac{2(1-2p)(1-p^{R_{\max}+1})}{A+B+C} \\ A = w \Big[ 1-(2p)^{m+1} \Big] (1-p) \\ B = (1-2p)(1-p^{R_{\max}+1}) \\ C = w \cdot 2^m p^{m+1} (1-2p)(1-p^{R_{\max}-m}) \\ p = 1-(1-\tau)^{n-1} \end{cases}$$
(19)

In the formula, A, B, and C are parameters. According to (18) and (19), the average access delay of the large-scale remote multi-user information management system is obtained.

$$E(D) = D_{\alpha} + \frac{1}{2} (T_c P_c + T_s P_s + t_{slov}) \begin{bmatrix} \frac{1 - (2p)^m}{1 - 2p} w - \frac{1 - p^m}{1 - p} - (2^m - 1)w p^{R_{\max}} + mp^{R_{\max}} \\ + (2^m w - 1) \cdot (\frac{p^m - p^{R_{\max}}}{1 - p}) - R_{\max} p^{R_{\max}} + mp^{R_{\max}} \end{bmatrix}$$
(20)

#### IV. CONCURRENCY CONTROL ALGORITHM

According to queuing delay and accessing delay in large-scale remote multi-user information management system, the concurrency control algorithm of the system is studied and the specific procedures are as following.

(1) The operation  $Op_i$  generated by the local site  $s_i$  of the system is immediately executed and go to the state modification module (6).

(2) When  $s_i$  receives the operation  $Op_j$  generated by remote node  $s_j$ , it needs to check its characteristic identification list. If there is a sign indicating that a new feature is added, it can be directly executed, and go to the state modification module (6). Otherwise, the local feature representation chain list and the remote operation feature representation list will be operated separately according to the characteristics. If there is a non-zero value, it means the operation object conflict, and go to step (3). If it has no conflict, this operation  $Op_j$  is performed on  $s_j$ , and go to the state modification module (6).

(3) The context  $ContextOp_i$  of the local site's state  $StateS_i$ and it is in the generation  $Op_j$  are compared. When the local station state and context satisfy (21), whether operation sequence or operation content is exactly the same, then execute  $Op_i$  and go to state modification module (6).

$$StateS_i = ContextOp_i$$
 (21)

(4) The context  $ContextOp_i$  of the local site's state  $StateS_i$ and it is in the generation  $Op_j$  are compared. If there is an operation  $Op_k \in ContextOp_j$ , and  $Op_k \notin StateS_i$ , the  $Op_j$  is hung up first and the algorithm is ended.

(5) c The context  $ContextOp_i$  of the local site's state  $StateS_i$ and it is in the generation  $Op_j$  are compared. If there is an operation  $Op_k \notin ContextOp_j$  and  $Op_k \in StateS_i$ , the priority of *Priority<sub>i</sub>* and *Priority<sub>j</sub>* of  $s_i$  and  $s_j$  are compared. When *Priority<sub>i</sub>* < *Priority<sub>j</sub>*, execute  $Op_j$  and go to step (6) or turn it off.

(6) In a large-scale remote multi-user information management system, each site maintains a feature identity chain list  $[Tag_{F_1}, Tag_{F_2}, \dots, Tag_{F_m}]$  and *m* is the number of features in the model. This operation has modified the feature object, and its identity  $Tag_{F_1}$  is 1. A new feature is added, and then a feature identification element is added to the end of the list, which is identified as 2. A feature is deleted, and its identity is 3. Which there is no change is identified as 0. Since each operation is mostly for an edited feature, only one element in the list of characteristic identities for each operation. The expression of  $L_i$  is:

$$L_i = \langle Priority_i, \Pr imaryFeature, ModifiedFeature, S_{intersect} \rangle$$
  
(22)

Among them,  $Priority_1$ , PrimaryFeature, *ModifiedFeature*,  $S_{intersect}$  are the results of each operation. The identification of the modified feature in the characteristic identifier chain is 1; The added feature identification is 2; And the deleted feature identification is 3. The algorithm ends.

$$StateS_i = StateS_i + [Op_i]$$
<sup>(23)</sup>

If all the information on the site are saved, it will cause a lot of redundancy and consume too much memory, and the efficiency will be reduced. Therefore, the operation  $Op_i$ , which is generated by site  $s_i$ , is executed on all cooperative sites, and the site can no longer save all the related information of  $Op_i$ .

$$L_i = L_i - \langle Priority_i, \Pr imaryFeature, ModifiedFeature, S_{intersect} \rangle$$
  
(24)

Since (22) is re-determined, the feature identified in the feature identification chain is restored, and the added feature is 0, and the element with feature identification 3 needs to be deleted from the linked list.

$$StateS_i = StateS_i - [Op_i]$$
<sup>(25)</sup>

# V. EXPERIMENTAL RESULTS AND ANALYSIS

In order to verify the overall effectiveness of the proposed concurrency control algorithm, simulation experiments are carried out. This experiment is based on the MATLAB platform, the data processor is NVIDIA GeForce GTX 1050 2G, the system memory is 64GB, and the access object is collaborative documents. Channel utilization is a measure of whether channel congestion occurs when the multi-user information management system is concurrently controlled, the performance index. In this paper, we adopt the concurrency control algorithm of large-scale remote multi-user information management system, the adaptive concurrency control algorithm based on collision rate prediction and the remote concurrency control algorithm based on sensor network, and the channel utilization and channel transmission efficiency of the three are compared. The formula for channel utilization and channel transmission efficiency is:

$$U = \frac{t}{t_0} \tag{26}$$

$$F = \frac{1 - P_c}{P_s} \tag{27}$$

Among them,  $t_0$  is the sending cycle.

The result of the test is shown in Fig. 3.



Fig. 3 Channel utilization of three different methods

Fig. 3 shows that, with the increase of concurrent information quantity and transmission path, the channel utilization of large-scale remote multi-user information management system is reduced from 99.9% to 98.3%, and that of the adaptive concurrency control algorithm based on the conflict rate prediction and the remote concurrency control algorithm based on the sensor network are reduced from 98.6% to 73.2% and 92.5% to 42.6%, respectively. Thus, with the increase of concurrent information quantity and transmission path, the channel utilization ratio of large-scale remote multi-user information management system is much higher than that of the other two algorithms. It indicates that the transmission capacity of each channel in large-scale remote multi-user information management system is high, and the channel is not jammed, which can complete concurrency control of multi-user information management system.

If you save all the information on the site, it will cause a lot of redundant information and consume too much memory. Three algorithms are used to test each other, and the information transmission efficiency of the three different methods is compared. The test results are shown in Fig. 4.



(c) Information transmission efficiency of remote concurrent control algorithm based on sensor network

Fig. 4 Information transmission efficiency of three different methods

Fig. 4 shows that the amount of data transmission increases with time, when the number of iterations is 3, the data output of the algorithm in this paper has reached 200bit, the data output of the adaptive concurrency control algorithm based on collision rate prediction is 150bit, and the data transmission of the remote concurrency control algorithm based on sensor network is 100bit, As the experiment progresses, and can be up to 350 bit and 260 bit by using the concurrency control algorithm of large-scale remote multi-user information management system and adaptive concurrency control algorithm based on collision rate prediction, respectively; While, that of the remote concurrency control algorithm based on sensor network has increased slowly, and the maximum is only 120 bit. After two iterations, the three algorithms have increased by 150 bit, 100 bit, and 20 bit respectively. The data output of the algorithm in this paper has the fastest growth. Therefore, compared with three different methods, we know that in the same time, the data transmission amount of concurrency control algorithm of large-scale remote multi-user information management system is much more than that of the other two algorithms, at the same time, the growth rate of the data output of the algorithm in this paper is also the fastest, which indicates that the information transmission efficiency of the algorithm is higher.

# VI. CONCLUSION

Concurrency control is the core of large-scale remote multi-user information management system. When the current concurrency control algorithm is used to control large-scale remote multi-user information management system, the transmission capacity of each channel is low, and the channel is prone to congestion. It cannot control the system accurately. In this paper, a concurrency control algorithm for large-scale remote multi-user information management system is presented. According to the average utilization rate of the system, the concurrency control structure and state of the system are analyzed and judged. Then the queuing delay and accessing delay of the system are calculated. Finally, the concurrent control of large-scale remote multi-user information management system is completed according to the delay of the system.

The experimental results show that the proposed method can complete concurrent control of large-scale remote multi-user information management system. However, there are still limitations, which can be optimized in the following aspects in future work.

(1) Every host has NIC and the file system needed to be allocated to the CPU and memory of the server. Its consumption is very huge, so energy consumption needs to be optimized.

(2) There are still some details needed to be optimized in the concurrency control structure of large-scale remote multi-user information management system, and it is necessary to carry out in-depth analysis and research.

(3) Repeated reading is not allowed. After reading the data, the update operation is performed, and the previous reading result cannot be reproduced.

#### References

- T. Wang, R. Johnson, and A. Fekete, "Efficiently making (almost) any concurrency control mechanism serializable," Vldb Journal, vol. 26, no. 4, pp. 1-26, 2017.
- [2] K. Barkaoui, and R. Benamara, "On concurrency control in mutlidatabase systems with an extended transaction model," Journal Supercomputing, vol. 24, pp. 193-202, 2003.
- [3] A. Ghosh, R. Chaki, and N. Chaki, "A new concurrency control mechanism for multi-threaded environment using transactional memory," Journal Supercomputing, vol. 71, no. 11, pp. 4095-4115, 2015.
- [4] C. Y. Oh, D. Hwang, and T. J. Lee, "Joint access control and resource allocation for concurrent and massive access of M2M devices," IEEE Transactions Wireless Communications, vol. 14, no. 8, pp. 4182-4192, 2015.
- [5] W. Remigiusz, A. Karatkevich, and M. Adamski,. "Prototyping of concurrent control systems with application of petri nets and comparability graphs," IEEE Transactions Control Systems Transactions, vol. 26, no. 2, pp. 575-586, 2018.
- [6] X. M. Zhang, Y. Zhang, and F. Yan, "Interference-based topology control algorithm for delay-constrained mobile ad hoc networks," IEEE Transactions Mobile Computer, vol. 14, no. 4, pp. 742-754, 2015.
- [7] B. J. Fan, and Y. Zhuang, "Adaptive concurrency control algorithm based on conflict-rate prediction," Computer Science, vol. 43, no. 11, pp. 280-283+290, 2016.
- [8] Z. H. Zeng, and C. Li, "Stack data structure of the embedded system scheduling control algorithm," Computer Simulation, vol. 33, no. 10, pp. 408-411, 2016.
- [9] W. F. Du, Z. Wu, and J. Q. Li, "A data distribution algorithm with buffer feedback in CMT over diversity paths," Journal of Chinese Computer Systems, vol. 36, no. 2, pp. 261-265, 2015.
- [10] M. L. Li, W. W. Cai, and X. Lv, "Parallel operational transformation algorithm in multi-core," Computer Science, vol. 43, no. 6, pp. 82-85, 2016.

- [11] W. Zhang, W. Liu, and X. Wang, "Online optimal generation control based on constrained distributed gradient algorithm," IEEE Transactions Power Systems, vol. 30, no. 1, pp. 35-45, 2015.
- [12] Y. Wang, J. Li, and X. Wang, "Research on pavement segmentation algorithm of high resolution remote sensing image," Journal of Computational Methods in Sciences and Engineering, vol. 19, no. S1, pp. S299-S305, 2019.
- [13] S. Duan, V. Shah-Mansouri, and Z. Wang, "D-ACB: Adaptive congestion control algorithm for bursty M2M traffic in LTE networks," IEEE Transactions Vehicle Technology, vol. 65, no. 12, pp. 9847-9861, 2016.
- [14] L. Huang, G. Huang, and R. P. Lebeau, "Optimization of Aifoil flow control using a genetic algorithm with diversity control," Journal Aircraft, vol. 44, no. 4, pp. 1337-1349, 2015.
- [15] M. Marinaki, Y. Marinakis, and G. E. Stavroulakis, "Fuzzy control optimized by a multi-objective differential evolution algorithm for vibration suppression of smart structures," Computer Structural, vol. 147, pp. 126-137, 2015.



Dawei Zhao, male, was born in February, 1981. His title is associate professor. He graduated from Beihua University in 2004, major in computer science and technology, with a bachelor degree. Now he is working in Network Center, Baicheng Vocational and Technical College. He mainly engages in the research of academic research direction and information management. So far, He has published 10 academic articles, participated in 4 scientific research topics and 1 software copyright patent.

# **Author Contribution**

Dawei Zhao: In this paper, a concurrency control algorithm for large-scale remote multi-user information management system is proposed. Through the analysis of the results, the delay of data link layer in multi-user information management system is carried out modeling. Experimental results show that the channel utilization rate of the proposed algorithm is over 98.3%, which can transmit large amounts of information in a relatively short time and concurrency control of information management system. He has made all his contributions to this paper.

# **Creative Commons Attribution License 4.0** (Attribution 4.0 International, CC BY 4.0)

This article is published under the terms of the Creative Commons Attribution License 4.0 <u>https://creativecommons.org/licenses/by/4.0/deed.en\_US</u>