

# **DATA TRANSMISSION METHOD AND SYSTEM EMPLOYING**

## **ASYNCHRONOUS RESPONSE**

### **TECHNICAL FIELD**

5   **[0001]** The present disclosure relates to the field of data transmission, in particular to a method and system for data transmission based on asynchronous response.

### **BACKGROUND**

10   **[0002]** In the process of data transmission from a data source to a data destination, an extract-transform-load (ETL) system is generally employed, which can extract the data, transform the format of the data, and load the data into the data destination. The conventional ETL system is suitable for frequent tasks that run on a fixed cycle (e.g., processing batch data) and meets low real-time requirements for data. Thus, the conventional ETL system is rarely applicable in scenarios having high real-time  
15 requirements for data.

**[0003]** The throughput (i.e. flowrate) of data transmission may need to be regulated during the process of data transmission. A conventional method for throughput regulation is to perform regulation by taking a system load of the data transmission system as a hard indicator. For example, the data transmission is disabled or the throughput thereof  
20 is decreased when the system load exceeds a certain threshold, and the throughput is restored when the system load starts to decrease. Another conventional method for throughput regulation is a method based on adaptive flow-limiting for protection, which takes the system load of the data transmission system as a reference value for the throughput control, in particular determines a currently allowable throughput from the  
25 current request response time and the request processing rate of the data transmission system, such that the throughput is increased as long as the data transmission system is not overloaded, without necessarily limiting the throughput below a certain threshold.

**[0004]** In order to maximize the throughput, it is also possible to apply an asynchronous response mode, where the module responsible for data loading in the data transmission  
30 system just transmits the data to the downstream data destination without waiting for a specific result of the data loading, and the data transmission system may process the specific result of the data loading by another module. In other words, for the data loaded in one cycle, it is not necessarily possible to learn in this cycle whether the data is loaded

successfully or not.

**[0005]** However, in the case of the asynchronous response mode, the aforesaid first method, which takes the system load of the data transmission system as the hard indicator, would limit the processing capacities of the data transmission system and the data destination and in turn decrease the data throughput of the data transmission system, because the data transmission is not allowed if the system load exceeds the threshold. Furthermore, when an event affecting the processing capacity, such as a failure or expansion, occurs at the downstream data destination, the method cannot detect the occurrence of the event to regulate the throughput in a timely manner, which may cause abnormal conditions of the data transmission system, such as idleness, massive data loss and the like. In addition, the aforesaid second method regulates the throughput according to the processing capacity of the data transmission system, and cannot dynamically control the throughput according to the processing capacity of the downstream data destination, which is therefore not applicable to a scenario where the processing capacity of the data destination is weaker than that of the data transmission system.

## **SUMMARY**

**[0006]** According to an aspect, the present disclosure provides a method for data transmission based on asynchronous response, which includes: an extraction step of extracting data from a first data terminal and storing the data in a cycle; a throughput regulation step of determining, based on a loading result fed back by a second data terminal in a previous cycle of said cycle and a first throughput of data transmission in the previous cycle, a second throughput for transmitting the data in said cycle; and a loading step of loading, in said cycle, the data into the second data terminal according to the second throughput.

**[0007]** According to another aspect, the present disclosure provides a method for data transmission based on asynchronous response, which includes: an extraction step of extracting a first data from a first data terminal and storing the first data into a first queue in a cycle; a throughput regulation step of determining, based on a first loading result fed back by a second data terminal in a previous cycle of said cycle and a first throughput of data transmission in the previous cycle, a second throughput for data transmission in said cycle; and a loading step of loading, in said cycle, the first data of the first queue and a second data of a second queue into the second data terminal according to the second throughput.

**[0008]** According to still another aspect, the present disclosure provides a system for data transmission based on asynchronous response, which includes: a data extraction module configured to extract data from a first data terminal and store the data in a cycle; a throughput regulation module configured to determine, based on a loading result fed back by a second data terminal in a previous cycle of said cycle and a first throughput of data transmission in the previous cycle, a second throughput for transmitting the data in said cycle; and a data loading module configured to load, in said cycle, the data into the second data terminal according to the second throughput.

**[0009]** According to yet another aspect, the present disclosure provides a system for data transmission based on asynchronous response, which includes: a data extraction module configured to extract a first data from a first data terminal and store the first data into a first queue in a cycle; a throughput regulation module configured to determine, based on a first loading result fed back by a second data terminal in a previous cycle of said cycle and a first throughput of data transmission in the previous cycle, a second throughput for data transmission in said cycle; and a data loading module configured to load, in said cycle, the first data of the first queue and a second data of a second queue into the second data terminal according to the second throughput.

**[0010]** According to still yet another aspect, the present disclosure provides a computer-readable medium having computer-readable instructions stored thereon, where the computer-readable instructions, when executed by a computer, can cause the method according to the embodiments of the present disclosure to be implemented.

**[0011]** According to the embodiments of the present disclosure, the throughput of the data transmission system can be automatically regulated based on the fed-back loading results in the case of asynchronous response, such that the throughput of the data transmission system can be increased as much as possible under the premise of ensuring that the data destination is not overwhelmed and reducing the data loss.

**[0012]** The aforesaid description is only a summary of the technical solutions of the present disclosure. In order to understand technical solutions of the present disclosure more clearly, the present disclosure may be implemented in accordance with the content described in the disclosure. In addition, specific embodiments of the present disclosure will be given below to make the above and other objects, features, and advantages of the present disclosure more obvious and understandable.

## **BRIEF DESCRIPTION OF THE DRAWINGS**

[0013] The aforesaid and other objects, features and advantages of exemplary embodiments of the present disclosure will be more readily understood upon reading the detailed description below with reference to the accompanying drawings. In the drawings, a certain number of embodiments of the present disclosure are illustrated in an exemplary rather than restrictive manner.

[0014] In the drawings, the same or corresponding reference signs indicate the same or corresponding portions.

[0015] FIG. 1 is a schematic diagram of a data transmission system according to some embodiments of the present disclosure.

[0016] FIG. 2 is a flowchart of a method for data extraction according to some embodiments of the present disclosure.

[0017] FIG. 3 is a timing diagram of a method for data loading according to some embodiments of the present disclosure.

[0018] FIG. 4 is a timing diagram of a method for throughput regulation according to some embodiments of the present disclosure.

[0019] FIG. 5 is a flowchart of a method for abnormal data handling according to some embodiments of the present disclosure.

## **DETAILED DESCRIPTION**

[0020] The principle and concept of the present disclosure will be described below with reference to a certain number of exemplary embodiments. It shall be understood that these embodiments are merely intended to enable those skilled in the art to better understand and thus implement the present disclosure, rather than limiting the scope of the present disclosure in any manner. To the contrary, these embodiments are provided for more thoroughly and completely understanding the present disclosure, and could more fully convey the scope of the present disclosure to those skilled in the art.

[0021] The specific embodiments of the present disclosure will be further illustrated in detail with reference to the drawings hereinafter.

[0022] FIG. 1 is a schematic diagram of a data transmission system according to some embodiments of the present disclosure.

[0023] As shown in FIG. 1, the system includes five modules: a data extraction module, a data loading module, an asynchronous response processing module, a cyclic scanning module, and a throughput regulation module. The system further includes a to-be-loaded

data queue, a to-be-reloaded data queue, and a to-be-acknowledged data Map.

**[0024]** The data extraction module may extract data from a data source, store the data in the to-be-loaded data queue after transforming the format of the data, and wait if the queue is full. In this disclosure, one data may represent one transaction, and each data may have a different size.

**[0025]** The data loading module may dequeue data from the to-be-loaded data queue and determine with the aid of the throughput regulation module whether the data can currently be loaded into the data destination. The data loading module may further include a normal data loading module and a failed data loading module, which are configured to load the data from the to-be-loaded data queue and from the to-be-reloaded data queue respectively. At each loading of data, the data loading module may simultaneously store the data in the to-be-acknowledged data Map. In certain embodiments, the to-be-acknowledged data Map is of a data storage structure for temporarily storing the data dumped from the data loading module. In the to-be-acknowledged data Map, the identity of the data is stored in the Key field, and the value of the data is stored in the Value field.

**[0026]** The asynchronous response processing module may process a processing result (i.e., a loading result of the data to be loaded, which may be success or failure) fed back from the data destination and count the amount of the data whose processing result is success or failure. For the data whose processing result is success, the asynchronous response processing module may remove said data directly from the to-be-acknowledged data Map. For the data whose processing result is failure, the asynchronous response processing module may transmit said data (i.e., that fails to be loaded) from the to-be-acknowledged data Map to the to-be-reloaded data queue, so as to be loaded by the data loading module. If the number of failures to load certain data exceeds the predetermined number (i.e., the maximum number of times the certain data can be reloaded), said data is discarded.

**[0027]** The cyclic scanning module may scan the to-be-acknowledged data Map to count the data whose processing result has not been received from the data destination within a certain period of time (i.e., timed-out data for which no feedback is received), and transmit said data from the to-be-acknowledged data Map to the to-be-reloaded data queue. The cyclic scanning module may further scan the processing result received from the data destination and stored in the asynchronous response processing module to calculate a success rate of data loading based on statistics in the current cycle and transmit the success rate to the throughput regulation module.

**[0028]** The throughput regulation module may regulate the transaction per second (TPS,

e.g., the number of transactions processed per second) of the data transmission system to achieve the throughput control. Firstly, a maximum TPS value in the most ideal case may be taken as the maximum throughput, and then the throughput (TPS value) of the data transmission system is increased or decreased based on the success rate of data loading in the current cycle to thereby achieve the throughput regulation for the system.

**[0029]** FIG. 2 shows a method for data extraction applicable to the data extraction module. As shown in FIG. 2, the method includes following steps.

**[0030]** (1) The data extraction module extracts data from the data source and transforms the data into a uniform format.

**[0031]** (2) If the to-be-loaded queue is not full currently, the transformed data is directly stored into the to-be-loaded queue, and then the process returns to step (1), otherwise proceeds to step (3).

**[0032]** (3) If the to-be-loaded queue is full currently, the number of failed transmissions is incremented by 1 for the current data, and if the number of failed transmissions does not exceed the maximum number of failed transmissions, the data extraction module re-executes step (2) after sleeping for a predetermined period of time, otherwise the data is discarded and the step (1) is re-executed.

**[0033]** FIG. 3 shows a timing diagram of a method for data loading applicable to the data loading module. As shown in FIG. 3, the data loading module may acquire the transformed data from the to-be-loaded data queue. After acquiring the data to be loaded, in order to ensure that the current throughput of the data transmission system does not exceed the allowable maximum throughput, the data loading module should send a request to the throughput regulation module for a token allowing the data to be loaded, so that the data to be loaded can be transmitted to the data destination only after the token is acquired.

**[0034]** When a request for a token is received, the throughput regulation module checks whether there is still a token available, and if there is no token available, the throughput regulation module may suspend the request and wait for a new token to be generated. The throughput regulation module may periodically generate or clear tokens. For example, at the end of each cycle, remaining tokens are cleared and a number of tokens corresponding to the maximum throughput currently allowed by the system are generated (e.g., if the maximum throughput is 1000, which for example indicates that up to 1000 data, e.g., 1000 transactions, can be processed per second, then the throughput regulation module may generate 1000 tokens to enable the processing of 1000 data).

The maximum throughput may be updated according to the data loading status at the

data destination. The throughput of the system may be regulated dynamically by modifying the number of tokens available in a cycle.

**[0035]** FIG. 4 is a timing diagram of a method for throughput regulation according to some embodiments of the present disclosure. The method involves the asynchronous response processing module, the cyclic scanning module, and the throughput regulation module. The method may include operations such as doing statistics on processing results from the data destination, counting the data for which no feedback is received before timing out, calculating the success rate of data loading in the current cycle, and updating the maximum throughput allowed by the system.

**[0036]** The asynchronous response processing module is configured to receive a loading result fed back from the data destination, process the corresponding data (i.e., remove the successfully loaded data from the to-be-acknowledged data Map, or transmit to the to-be-reloaded data queue the data failing to be loaded or the timed-out data for which no feedback is received), and update a value of a loading success counter or a loading failure counter based on the loading result fed back.

**[0037]** The cyclic scanning module may perform the scanning in each cycle and calculate the success rate of data loading into the data destination in the current cycle (for the data supposed to be loaded into the data destination in the previous cycle and may be referred to as historical data), so as to trigger the throughput regulation module to regulate the throughput. For calculating the success rate, the cyclic scanning module may firstly count the amount of all the timed-out data for which no feedback is received in the current cycle (stored in a timed-out data counter, for example), and then acquire current values of the loading success counter and the loading failure counter to calculate the success rate according to the following formula:

$$S = \frac{C_s}{C_s + C_f + C_t} * 100\%$$

**[0038]** where S is the success rate of data loading for the current cycle,  $C_s$  and  $C_f$  are the current values of the loading success counter and the loading failure counter respectively, and  $C_t$  is a current value of the timed-out data counter. The value of each counter is cleared immediately after calculating the S to ensure that there is no interference with the calculation in the next cycle.

**[0039]** The method of throughput regulation may be summarized as follows.

**[0040]** (1) An acceptable maximum throughput  $T_{max}$  is configured according to the processing capacity of the data destination.

**[0041]** (2) If the success rate of data loading based on the feedbacks received from the data destination in the current cycle is lower than an acceptable minimum success rate, the throughput of the data transmission system may be decreased based on a fixed step; if the throughput is lower than the minimum throughput, the minimum throughput may be taken as the actual throughput (the number of corresponding tokens generated according to the determined throughput value); and if the throughput before the decrease has reached the acceptable minimum throughput, the throughput of the system is no longer decreased and manual intervention is required for troubleshooting.

**[0042]** (3) If the success rate of data loading for the current cycle is greater than the success rate regulation threshold (which is greater than the minimum success rate) and the current throughput of the system is less than  $T_{\max}$ , the throughput may be increased according to the fixed step; if the throughput is greater than the maximum throughput, the maximum throughput may be taken as the actual throughput (the number of corresponding tokens generated according to the determined throughput value); and if the throughput before the increase has reached  $T_{\max}$ , the throughput value of the system is no longer increased.

**[0043]** (4) If the success rate is greater than the minimum success rate and less than the success rate regulation threshold and the throughput is less than  $T_{\max}$ , the data transmission system and the downstream system (i.e., the data destination) are both considered to be in a stable state, which requires no regulation for the throughput of the data transmission system.

**[0044]** (5) If the throughput of the data transmission system is maintained at  $T_{\max}$  and the success rate as calculated is always greater than or equal to the success rate regulation threshold, an attempt may be made to increase  $T_{\max}$ ; and if the success rate decreases immediately after increasing  $T_{\max}$ , the previous  $T_{\max}$  is restored and no more attempt may be made to increase  $T_{\max}$  within a certain period (e.g., on the same day).

**[0045]** (6) If the data destination is expanded,  $T_{\max}$  may be manually modified to the maximum throughput that is compatible with the expanded processing capacity.

**[0046]** (7) If after the data transmission system increases the throughput because the success rate as calculated is greater than or equal to the success rate regulation threshold, the success rate as calculated decreases below the minimum success rate such that the data transmission system decreases the throughput and the process of first increasing and then decreasing the throughput is further repeated several times, it may indicate that the calculated  $T_{\max}$  does not meet the current status, which thereby requires the manual intervention for troubleshooting (e.g., decreasing the  $T_{\max}$  manually).

**[0047]** An embodiment for realizing the method for throughput regulation will be described below. Firstly, the parameters are defined, and their meanings are as follows.

**[0048]**  $T_{\max}$  refers to a maximum TPS (i.e., the maximum throughput) bearable at the data destination under ideal conditions, and the  $T_{\max}$  can be manually configured and modified by the system operation and maintenance personnel.

**[0049]**  $T_{\min}$  refers to a minimum acceptable TPS at the data destination, and if the value is below  $T_{\min}$ , the manual intervention is required for troubleshooting.

**[0050]**  $S_{\min}$  refers to a minimum success rate of data loading allowed at the data destination.

**[0051]**  $S_{\text{adp}}$  refers to a minimum acceptable success rate (i.e. success rate regulation threshold) for increasing the TPS, and the TPS can be increased only when the current success rate is greater than or equal to  $S_{\text{adp}}$ .

**[0052]**  $st$  refers to a step size for the TPS regulation, i.e. the increased or decreased value of the TPS each time.

**[0053]**  $T_{\text{cur}}$  refers to a TPS value currently allowed at the data destination, and  $T_{\text{cur}}$  may be configured arbitrarily so that  $T_{\text{cur}}$  is not greater than  $T_{\max}$  when starting the data transmission system.

**[0054]** In addition, in this disclosure, a continuous increase-decrease of the TPS or a continuous decrease-increase of the TPS is defined as an oscillation process. If the oscillation occurs consecutively, it means that the  $T_{\max}$  value does not meet the current status of the data destination, and the operation and maintenance personnel shall be reminded to perform troubleshooting and confirmation and may modify the  $T_{\max}$  value if necessary.

**[0055]** Based on this, each regulation operation may be recorded, for example, with a flag bit UD.

**[0056]** If UD is 00, it indicates that TPS is unregulated; if UD is 10, it indicates that TPS is increased; and if UD is 01, it indicates that TPS is decreased.

**[0057]** Furthermore,  $UD_p$  is used to record the last throughput regulation operation. For example,  $UD|UD_p$  may be calculated (e.g., by operation) to indicate whether the oscillation occurs (e.g. if  $UD|UD_p$  is 11, it indicates that an oscillation occurs).  $C_{\text{con}}$  is used to record the number of consecutive oscillations, and  $M_{\text{con}}$  is used to indicate an acceptable maximum number of consecutive oscillations.

**[0058]** Based on the aforesaid definitions, the method for throughput regulation is implemented as follows.

**[0059]** When  $S \geq S_{\text{adp}}$  and  $T_{\text{cur}} < T_{\max}$ , UD is 10. Then,  $T_{\text{cur}}$  is updated by  $T_{\text{cur}} += st$  and

UD|UD<sub>p</sub> is calculated;

C<sub>con</sub> is modified by C<sub>con</sub>+=1 and UD<sub>p</sub> is modified to 10 if UD|UD<sub>p</sub> is 11; and

C<sub>con</sub> is modified to 0 and UD<sub>p</sub> is modified to 10 if UD|UD<sub>p</sub>! is 11.

**[0060]** When  $S \geq S_{adp}$  and  $T_{cur} = T_{max}$ ,  $T_{cur}$  remains unchanged, C<sub>con</sub> is modified to 0.

5 and UD<sub>p</sub> is modified to 00.

**[0061]** When  $S \leq S_{min}$  and  $T_{cur} > T_{min}$ , UD is 01. Then,  $T_{cur}$  is updated by  $T_{cur}=st$ , and UD|UD<sub>p</sub> is calculated;

C<sub>con</sub> is modified by C<sub>con</sub>+=1 and UD<sub>p</sub> is modified to 01 if UD|UD<sub>p</sub> is 11; and

C<sub>con</sub> is modified to 0 and UD<sub>p</sub> is modified to 01 if UD|UD<sub>p</sub>! is 11.

10 **[0062]** When  $S \leq S_{min}$  and  $T_{cur} = T_{min}$ ,  $T_{cur}$  remains unchanged, C<sub>con</sub> is modified to 0, and UD<sub>p</sub> is modified to 00; in this case, an alarm shall be issued to prompt manual intervention for troubleshooting.

**[0063]** When  $S_{min} < S < S_{adp}$ ,  $T_{cur}$  remains unchanged, C<sub>con</sub> is modified to 0, and UD<sub>p</sub> is modified to 00.

15 **[0064]** Each time the algorithm for the throughput regulation is executed, a corresponding number of tokens are firstly generated based on the updated TPS value allowed at the data destination, and C<sub>con</sub> is checked simultaneously. If  $C_{con} \geq M_{con}$ , it means that the number of consecutive oscillations of the data transmission system exceeds the maximum value, that is, the value of  $T_{max}$  is not suitable for the current data destination. Thus, the data transmission system may issue an alarm to remind the operation and maintenance personnel to intervene for troubleshooting.

20 **[0065]** In order to reduce the abnormal data (which includes the data failing to be loaded or the timed-out data for which no feedback is received), the asynchronous response processing module or the cyclic scanning module may transmit the abnormal data to the to-be-reloaded data queue and reload the abnormal data via a failure data reloading module in the data loading module. FIG. 5 is a flowchart of a method for handling abnormal data according to some embodiments of the present disclosure. The method for handling abnormal data includes following steps.

25 **[0066]** (1) The asynchronous response processing module performs a determination after receiving the processing result, and if the result is success, the corresponding data is directly removed from the to-be-acknowledged data Map and the process ends; otherwise, the process proceeds to step (2).

30 **[0067]** (2) If the result is failure, the number of failures is updated; and if the number of failures exceeds the maximum value, the data is directly removed from the to-be-acknowledged data Map and an abnormality is reported (e.g., issuing an alarm);

35

otherwise the process proceeds to step (3).

**[0068]** (3) If the to-be-reloaded data queue is not full currently, the data may be directly dumped to the to-be-reloaded data queue; otherwise, it is regarded as a dumping failure and the number of failures thereof is directly updated, and the process proceeds to step (4).

**[0069]** (4) If the number of data failures exceeds the maximum number of failures, the data is directly discarded and an abnormality is reported; otherwise, the data continues to be stored in the to-be-acknowledged data Map, and the process returns to step (3) after waiting for a period of time.

**[0070]** As shown in FIG. 1 and FIG. 5, the method for handling abnormal data may reduce the impact of abnormal data on the system by loading segregation. In other words, a normal data loading module is configured to process the data in the to-be-loaded data queue, and a failed data loading module is configured to process the data in the to-be-reloaded data queue, the two modules performing the operation in parallel. Based on this, the TPS value  $T_{cur}$  allowed for the current system or data destination, which is calculated by the throughput regulation module, may be subdivided into normal data loading TPS (noted as  $T_{nor}$ ) and abnormal data loading TPS (noted as  $T_{uno}$ ), and different loading modules may request their respective loading tokens to thereby achieve the loading segregation between normal and abnormal data. The specific method includes following steps.

**[0071]** (1) If the success rate is not 100%, the success rate is multiplied by the current throughput to be taken as the throughput for the normal data loading, and the remaining part of the current throughput is taken as the throughput for the abnormal data loading. Then, the percentage of the throughput for abnormal data in the current throughput is updated to be available for the data transmission system in subsequent cycles. The specific implementation method may for example be as follows:

**[0072]** When  $S \neq 100\%$ ,  $T_{nor} = T_{cur} * S$ ,  $T_{uno} = T_{cur} * (1 - S)$ ; and  $R_p$  is updated as  $T_{uno}/T_{cur}$ .

**[0073]** (2) If the success rate is 100%, the throughput for the abnormal data loading is the minimum value between the amount of data in the to-be-reloaded data queue and a product of the aforesaid percentage and the current throughput. Thus, the throughput for the normal data loading is a difference between the current throughput and the throughput for the abnormal data loading. The percentage of the throughput for the abnormal data in the current throughput may be updated to be available for the data transmission system

in subsequent cycles. The specific implementation method may for example be as follows:

**[0074]** When  $S = 100\%$ ,  $T_{\text{uno}} = \min(L_u, R_p * T_{\text{cur}})$ ,  $T_{\text{uno}} = T_{\text{cur}} * (1 - S)$ ; and  $R_p$  is updated as  $T_{\text{uno}}/T_{\text{cur}}$ .

**[0075]** where  $S$  is the success rate of the data loading in the current cycle,  $R_p$  is the percentage of abnormal data loading TPS in the total TPS of the system in the previous cycle, and  $L_u$  is the amount of data in the to-be-reloaded data queue (e.g., the number of transactions).

**[0076]** The program code for performing operations of the present disclosure can be written in any combination of one or more programming languages, and the programming languages include object-oriented programming languages, such as Java, C++, etc., and further include conventional procedural programming languages, such as "C " language or similar programming languages. The program code may be executed entirely on the user computing device, partially on the user device, partially on the user computing device, or entirely on a remote computing device or server. In the case of involving a remote computing device, the remote computing device may be connected to the user computing device over any kind of network which includes a local area network (LAN) or wide area network (WAN), or may be connected to an external computing device (e.g., via Internet provided from an Internet service provider).

**[0077]** Furthermore, although the operations of the method of the present disclosure are described in a specific order in drawings, it does not require or imply that the operations must be performed in the specific order, or that the desired result can be achieved only if all the operations as shown are performed. Additionally or alternatively, some steps may be omitted, multiple steps may be combined into one step for execution, and/or one step may be decomposed into multiple steps for execution.

**[0078]** It shall be noted that although several software devices/modules and sub-devices/modules implementing the aforesaid method are mentioned in the above detailed description, such a division is not mandatory. In fact, according to embodiments of the present disclosure, the features and functions of two or more devices described above may be embodied in a single device/module. Conversely, the features and functions of a single device/module described above may be further divided as being embodied by a plurality of devices/modules.

**[0079]** Although the concepts and principles of the present disclosure have been described with reference to several embodiments, it shall be understood that the present disclosure is not limited to the embodiments as disclosed, and the division of various

aspects does not imply that the features in those aspects cannot be combined for benefit; instead, such division is merely for convenience of presentation. The present disclosure is intended to cover various modifications and equivalent arrangements included within the concept and scope of the appended claims.

## WHAT IS CLAIMED IS:

1. A method for data transmission based on asynchronous response, comprising:

5 an extraction step of both extracting data from a first data terminal and storing the data in a cycle;

a throughput regulation step of determining, based on a loading result fed back by a second data terminal in a previous cycle of said cycle and a first throughput of data transmission in the previous cycle, a second throughput for transmitting the data in said cycle; and

10 a loading step of loading, in said cycle, the data into the second data terminal according to the second throughput,

wherein the loading result comprises the amount of historical data for which loading success is fed back by the second data terminal in the previous cycle and the amount of historical data for which loading failure is fed back by the second data terminal in the  
15 previous cycle,

wherein the throughput regulation step further comprises:

counting the amount of timed-out historical data for which no feedback is received from the second data terminal within a predetermined period of time in the previous cycle, and

20 calculating a success rate of loading historical data based on the feedback received from the second data terminal in the previous cycle, the success rate being used to determine the second throughput and calculated by a following formula:

$$S = \frac{C_s}{C_s + C_f + C_t} * 100\%$$

25 wherein S is the success rate,  $C_s$  is the amount of historical data for which loading success is fed back by the second data terminal in the previous cycle,  $C_f$  is the amount of historical data for which loading failure is fed back by the second data terminal in the previous cycle, and  $C_t$  is the amount of time-out historical data for which no feedback is received from the second data terminal within the predetermined period of time in the previous cycle.

2. The method as claimed in claim 1, wherein the extraction step comprises:

(a) extracting the data from the first data terminal and transforming the data into a predetermined format;

5 (b) determining whether a queue is full, and if the queue is not full, storing the transformed data into the queue and returning to step (a); otherwise, proceeding to step (c); and

10 (c) incrementing the number of failed transmissions by 1 for the transformed data, and if the number of failed transmissions does not exceed a maximum number of failed transmissions, returning to the step (b) after waiting for a predetermined period of time; otherwise, discarding the transformed data and returning to the step (a).

3. The method as claimed in claim 1, wherein the throughput regulation step comprises:

15 (a) decreasing the first throughput by a first predetermined value to acquire the second throughput if the success rate is less than or equal to a minimum success rate and the first throughput is greater than a minimum throughput, and taking the minimum throughput as the second throughput if the decreased first throughput is less than the minimum throughput;

(b) taking the minimum throughput as the second throughput if the success rate is less than or equal to the minimum success rate and the first throughput is equal to the minimum throughput;

20 (c) increasing the first throughput by a second predetermined value to acquire the second throughput if the success rate is greater than a success rate regulation threshold and the first throughput is less than a maximum throughput, and taking the maximum throughput as the second throughput if the increased first throughput is greater than the maximum throughput;

25 (d) taking the maximum throughput as the second throughput if the success rate is greater than the success rate regulation threshold and the first throughput reaches the maximum throughput; or

(e) taking the first throughput as the second throughput if the success rate falls between the minimum success rate and the success rate regulation threshold.

30 4. The method as claimed in claim 3, wherein the throughput regulation step further comprises:

increasing the maximum throughput and the second throughput by the second predetermined value when the second throughput remains at the maximum throughput and the success rate is greater than the success rate regulation threshold for a first predetermined number of cycles.

- 5 5. The method as claimed in claim 4, wherein if the success rate decreases in a next cycle of said cycle after increasing the maximum throughput and the second throughput by the second predetermined value, the increased maximum throughput and the increased second throughput are restored to the maximum throughput and the second throughput prior to the increasing and the maximum throughput is no longer increased  
10 for a second predetermined number of cycles.

6. The method as claimed in claim 3, further comprising:

determining that the maximum throughput is set incorrectly and issuing an alarm message if the following operations are repeated in a plurality of cycles for a preset number of times:

- 15 performing the step (a) directly after performing the step (c), or  
performing the step (c) directly after performing the step (a).

7. The method as claimed in claim 1, wherein the loading step comprises:

generating a token for loading the data in said cycle based on the second throughput and the success rate;

- 20 extracting data in the queue based on the token; and  
loading the data extracted from the queue into the second data terminal.

8. A method for data transmission based on asynchronous response, comprising:

an extraction step of both extracting first data from a first data terminal and storing the first data into a first queue in a cycle;

- 25 a throughput regulation step of determining, based on a first loading result fed back by a second data terminal in a previous cycle of said cycle and a first throughput of data transmission in the previous cycle, a second throughput for data transmission in said cycle; and

- a loading step of loading, in said cycle, the first data of the first queue and second  
30 data of a second queue into the second data terminal according to the second throughput,

wherein the first loading result comprises the amount of historical data for which loading success is fed back by the second data terminal in the previous cycle and the amount of historical data for which loading failure is fed back by the second data terminal in the previous cycle,

5 wherein the throughput regulation step comprises:

counting the amount of timed-out historical data for which no feedback is received from the second data terminal within a predetermined period of time in the previous cycle, and

calculating a success rate of loading historical data based on the feedback received  
10 from the second data terminal in the previous cycle, the success rate being used to determine the second throughput and calculated by a following formula:

$$S = \frac{C_s}{C_s + C_f + C_t} * 100\%$$

wherein S is the success rate,  $C_s$  is the amount of historical data for which loading success is fed back by the second data terminal in the previous cycle,  $C_f$  is the amount  
15 of historical data for which loading failure is fed back by the second data terminal in the previous cycle, and  $C_t$  is the amount of timed-out historical data for which no feedback is received from the second data terminal within the predetermined period of time in the previous cycle,

wherein the second data refers to data which has failed to be loaded or for which no  
20 feedback has been received from the second data terminal in said cycle.

9. The method as claimed in claim 8, further comprising:

storing the first data and the second data into a data structure as candidate data.

10. The method as claimed in claim 9, further comprising:

determining whether the candidate data in the data structure is successfully loaded  
25 based on a second loading result fed back by the second data terminal in said cycle or at least one cycle after said cycle; if the candidate data is successfully loaded, deleting the candidate data in the data structure; if the candidate data fails to be loaded, determining whether the candidate data has been loaded more than a predetermined number of times; deleting the candidate data in the data structure if the predetermined

number of times is exceeded; and transmitting the candidate data to the second queue as the second data if the predetermined number of times is not exceeded.

11. The method as claimed in claim 8, wherein the extraction step comprises:

5 (a) extracting the first data from the first data terminal and transforming the first data into a predetermined format;

(b) determining whether the first queue is full, and if the first queue is not full, storing the transformed first data into the first queue and returning to step (a); otherwise, proceeding to step (c); and

10 (c) incrementing the number of failed transmissions by 1 for the transformed first data, and if the number of failed transmissions does not exceed a maximum number of failed transmissions, returning to step (b) after waiting for a predetermined period of time; otherwise, discarding the transformed data and returning to the step (a).

12. The method as claimed in claim 8, wherein the throughput regulation step comprises:

15 (a) decreasing the first throughput by a first predetermined value to acquire the second throughput if the success rate is less than or equal to a minimum success rate and the first throughput is greater than a minimum throughput, and taking the minimum throughput as the second throughput if the decreased first throughput is less than the minimum throughput;

20 (b) taking the minimum throughput as the second throughput if the success rate is less than or equal to the minimum success rate and the first throughput is equal to the minimum throughput;

25 (c) increasing the first throughput by a second predetermined value to acquire the second throughput if the success rate is greater than a success rate regulation threshold and the first throughput is less than a maximum throughput, and taking the maximum throughput as the second throughput if the increased first throughput is greater than the maximum throughput;

(d) taking the maximum throughput as the second throughput if the success rate is greater than the success rate regulation threshold and the first throughput reaches the maximum throughput; or

30 (e) taking the first throughput as the second throughput if the success rate falls between the minimum success rate and the success rate regulation threshold.

13. The method as claimed in claim 12, wherein the throughput regulation step further comprises:

5 increasing the maximum throughput and the second throughput by the second predetermined value when the second throughput remains at the maximum throughput and the success rate is greater than the success rate regulation threshold for a first predetermined number of cycles.

14. The method as claimed in claim 13, wherein if the success rate decreases in a next cycle of said cycle after increasing the maximum throughput and the second throughput by the second predetermined value, the increased maximum throughput and the increased second throughput are restored to the maximum throughput and the second throughput prior to the increasing and the maximum throughput is no longer increased for a second predetermined number of cycles.

15. The method as claimed in claim 12, further comprising:

15 determining that the maximum throughput is set incorrectly and issuing an alarm message if the following operations are repeated in a plurality of cycles for a preset number of times:

performing the step (a) directly after performing the step (c), or

performing the step (c) directly after performing the step (a).

16. The method as claimed in claim 8, wherein the loading step comprises:

20 (a) generating in said cycle a first token for loading the first data and a second token for loading the second data based on the second throughput and the success rate;

(b) extracting the first data in the first queue based on the first token and extracting the second data in the second queue based on the second token; and

25 (c) loading the extracted first data and the extracted second data into the second data terminal.

17. The method as claimed in claim 16, wherein the step (a) comprises:

when the success rate is less than 100%, obtaining a throughput for loading the first data by multiplying the success rate with the second throughput, obtaining a throughput for loading the second data by subtracting the throughput for loading the first data from the second throughput, updating a ratio of the throughput for loading the second data to

the second throughput, and generating the first token based on the throughput for loading the first data and generating the second token based on the throughput for loading the second data; or

when the success rate is equal to 100%, taking, as the throughput for loading the second data, the minimum value between the amount of data in the second queue and a product of said ratio and the second throughput, obtaining a throughput for loading the first data by subtracting the throughput for loading the second data from the second throughput, updating said ratio, and generating the first token based on the throughput for loading the first data and generating the second token based on the second token for loading the second data.

18. A computer-readable medium having computer-readable instructions stored thereon, wherein the computer-readable instructions, when executed by a computer, can cause the method as claimed in any one of claims 1-17 to be implemented.

19. A system for data transmission based on asynchronous response, comprising:

a data extraction module configured to both extract data from a first data terminal and store the data in a cycle;

a throughput regulation module configured to determine, based on a loading result fed back by a second data terminal in a previous cycle of said cycle and a first throughput of data transmission in the previous cycle, a second throughput for transmitting the data in said cycle; and

a data loading module configured to load, in said cycle, the data into the second data terminal according to the second throughput,

wherein the loading result comprises the amount of historical data for which loading success is fed back by the second data terminal in the previous cycle and the amount of historical data for which loading failure is fed back by the second data terminal in the previous cycle,

wherein the system further comprises a cyclic scanning module configured to: count the amount of timed-out historical data for which no feedback is received from the second data terminal within a predetermined period of time in the previous cycle, calculate a success rate of loading historical data based on the feedback received from the second data terminal in the previous cycle, and transmit the success rate to the throughput regulation module, wherein the success rate is calculated by a following formula:

$$S = \frac{C_s}{C_s + C_f + C_t} * 100\%$$

wherein S is the success rate,  $C_s$  is the amount of historical data for which loading success is fed back by the second data terminal in the previous cycle,  $C_f$  is the amount of historical data for which loading failure is fed back by the second data terminal in the previous cycle, and  $C_t$  is the amount of timed-out historical data for which no feedback is received from the second data terminal within the predetermined period of time in the previous cycle.

20. The system as claimed in claim 19, wherein the data extraction module is further configured to execute following operations:

(a) extracting the data from the first data terminal and transforming the data into a predetermined format;

(b) determining whether a queue is full, and if the queue is not full, storing the transformed data into the queue and returning to step (a); otherwise, proceeding to step (c); and

(c) incrementing the number of failed transmissions by 1 for the transformed data, and if the number of failed transmissions does not exceed a maximum number of failed transmissions, returning to step (b) after waiting for a predetermined period of time; otherwise, discarding the transformed data and returning to the step (a).

21. The system as claimed in claim 19, wherein the throughput regulation module further comprises:

a first module configured to decrease the first throughput by a first predetermined value to acquire the second throughput if the success rate is less than or equal to a minimum success rate and the first throughput is greater than a minimum throughput, and take the minimum throughput as the second throughput if the decreased first throughput is less than the minimum throughput;

a second module configured to take the minimum throughput as the second throughput if the success rate is less than or equal to the minimum success rate and the first throughput is equal to the minimum throughput;

a third module configured to increase the first throughput by a second predetermined value to acquire the second throughput if the success rate is greater than

a success rate regulation threshold and the first throughput is less than a maximum throughput, and take the maximum throughput as the second throughput if the increased first throughput is greater than the maximum throughput;

5 a fourth module configured to take the maximum throughput as the second throughput if the success rate is greater than the success rate regulation threshold and the first throughput reaches the maximum throughput; and

a fifth module configured to take the first throughput as the second throughput if the success rate falls between the minimum success rate and the success rate regulation threshold.

10 22. The system as claimed in claim 21, wherein the throughput regulation module further comprises:

15 a sixth module configured to increase the maximum throughput and the second throughput by the second predetermined value when the second throughput remains at the maximum throughput and the success rate is greater than the success rate regulation threshold for a first predetermined number of cycles.

20 23. The system as claimed in claim 22, wherein after increasing the maximum throughput and the second throughput by the second predetermined value, the throughput regulation module is further configured to, if the success rate decreases in a next cycle of said cycle, restore the increased maximum throughput and the increased second throughput to the maximum throughput and the second throughput prior to the increasing and no longer increase the maximum throughput for a second predetermined number of cycles.

25 24. The system as claimed in claim 21, wherein the throughput regulation module is further configured to determine that the maximum throughput is set incorrectly and issue an alarm message if the following operations are repeated in a plurality of cycles for a preset number of times:

executing the first module directly after executing the third module; or

executing the third module directly after executing the first module.

25 25. The system as claimed in claim 19, wherein

30 the throughput regulation module further comprises:

a module configured to generate a token for loading the data in said cycle based on

the second throughput and the success rate;

the data loading module further comprises:

a module configured to extract data in the queue based on the token; and

5 a module configured to load the data extracted from the queue into the second data terminal.

26. A system for data transmission based on asynchronous response, comprising:

a data extraction module configured to both extract first data from a first data terminal and store the first data into a first queue in a cycle;

10 a throughput regulation module configured to determine, based on a first loading result fed back by a second data terminal in a previous cycle of said cycle and a first throughput of data transmission in the previous cycle, a second throughput for data transmission in said cycle; and

15 a data loading module configured to load, in said cycle, the first data of the first queue and second data of a second queue into the second data terminal according to the second throughput,

wherein the first loading result comprises the amount of historical data for which loading success is fed back by the second data terminal in the previous cycle and the amount of historical data for which loading failure is fed back by the second data terminal in the previous cycle,

20 wherein the system further comprises a cyclic scanning module configured to count the amount of timed-out historical data for which no feedback is received from the second data terminal within a predetermined period of time in the previous cycle, calculate a success rate of loading historical data based on the feedback received from the second data terminal in the previous cycle, and transmit the success rate to the throughput  
25 regulation module, wherein the success rate is calculated by a following formula:

$$S = \frac{C_s}{C_s + C_f + C_t} * 100\%$$

wherein S is the success rate,  $C_s$  is the amount of historical data for which loading success is fed back by the second data terminal in the previous cycle,  $C_f$  is the amount of historical data for which loading failure is fed back by the second data terminal in the

previous cycle, and  $C_t$  is the amount of timed-out historical data for which no feedback is received from the second data terminal within the predetermined period of time in the previous cycle,

5 wherein the second data refers to data which has failed to be loaded or for which no feedback has been received from the second data terminal in said cycle.

27. The system as claimed in claim 26, wherein the data loading module is further configured to store the first data and the second data into a data structure as candidate data.

28. The system as claimed in claim 27, further comprising:

10 an asynchronous response processing module configured to: determine whether the candidate data in the data structure is successfully loaded based on a second loading result fed back by the second data terminal in said cycle or at least one cycle after said cycle; if the candidate data is successfully loaded, delete the candidate data in the data structure; if the candidate data fails to be loaded, determine whether the candidate data  
15 has been loaded more than a predetermined number of times; delete the candidate data in the data structure if the predetermined number of times is exceeded, and transmit the candidate data to the second queue as the second data if the predetermined number of times is not exceeded.

29. The system as claimed in claim 26, wherein the data extraction module is further  
20 configured to execute following operations:

(a) extracting the first data from the first data terminal and transforming the first data into a predetermined format;

(b) determining whether the first queue is full, and storing the transformed first data into the first queue and returning to step (a) if the first queue is not full; otherwise,  
25 proceeding to step (c); and

(c) incrementing the number of failed transmissions by 1 for the transformed first data, and if the number of failed transmissions does not exceed a maximum number of failed transmissions, returning to step (b) after waiting for a predetermined period of time; otherwise, discarding the transformed data and returning to the step (a).

30 30. The system as claimed in claim 26, wherein the throughput regulation module further comprises:

a first module configured to decrease the first throughput by a first predetermined value to acquire the second throughput if the success rate is less than or equal to a minimum success rate and the first throughput is greater than a minimum throughput, and take the minimum throughput as the second throughput if the decreased first throughput is less than the minimum throughput;

a second module configured to take the minimum throughput as the second throughput if the success rate is less than or equal to the minimum success rate and the first throughput is equal to the minimum throughput;

a third module configured to increase the first throughput by a second predetermined value to acquire the second throughput if the success rate is greater than a success rate regulation threshold and the first throughput is less than a maximum throughput, and take the maximum throughput as the second throughput if the increased first throughput is greater than the maximum throughput;

a fourth module configured to take the maximum throughput as the second throughput if the success rate is greater than the success rate regulation threshold and the first throughput reaches the maximum throughput; and

a fifth module configured to take the first throughput as the second throughput if the success rate falls between the minimum success rate and the success rate regulation threshold.

31. The system as claimed in claim 30, wherein the throughput regulation module further comprises:

a sixth module configured to increase the maximum throughput and the second throughput by the second predetermined value when the second throughput remains at the maximum throughput and the success rate is greater than the success rate regulation threshold for a first predetermined number of cycles.

32. The system as claimed in claim 31, wherein after increasing the maximum throughput and the second throughput by the second predetermined value, the throughput regulation module is further configured to, if the success rate decreases in a next cycle of said cycle, restore the increased maximum throughput and the increased second throughput to the maximum throughput and the second throughput prior to the increasing and no longer increase the maximum throughput for a second predetermined number of cycles.

33. The system as claimed in claim 30, wherein the throughput regulation module is further configured to determine that the maximum throughput is set incorrectly and issue an alarm message if the following operations are repeated in a plurality of cycles for a preset number of times:

- 5       executing the first module directly after executing the third module; or
- executing the third module directly after executing the first module.

34. The system as claimed in claim 26, wherein

the throughput regulation module further comprises:

- 10       a module configured to generate a first token for loading the first data and a second token for loading the second data in said cycle based on the second throughput and the success rate;

the data loading module further comprises:

- 15       a module configured to extract the first data in the first queue based on the first token and extract the second data in the second queue based on the second token; and
- a module configured to load the extracted first data and the extracted second data into the second data terminal.

35. The system as claimed in claim 34, wherein the module configured to generate the first token for loading the first data and the second token for loading the second data in said cycle based on the second throughput and the success rate further comprises:

- 20       a module configured to, when the success rate is less than 100%, obtain a throughput for loading the first data by multiplying the success rate with the second throughput, obtain a throughput for loading the second data by subtracting the throughput for loading the first data from the second throughput, update a ratio of the throughput for loading the second data to the second throughput, and generate the first
- 25       token based on the throughput for loading the first data and generate the second token based on the throughput for loading the second data; or

- a module configured to, when the success rate is equal to 100%, take, as the throughput for loading the second data, the minimum value between the amount of data in the second queue and a product of said ratio and the second throughput, obtain a
- 30       throughput for loading the first data by subtracting the throughput for loading the second data from the second throughput, update said ratio, and generate the first token based

on the throughput for loading the first data and generate the second token based on the second token for loading the second data.