

METHOD AND APPARATUS FOR UPLOADING STREAMING DATA, ACCESS DEVICE AND STORAGE MEDIUM

TECHNICAL FIELD

[0001] Embodiments of the present disclosure relate to the field of computer technologies, and in particular to a method and apparatus for uploading streaming data, an access device and a storage medium.

BACKGROUND

[0002] Data uploading is configured to upload streaming data acquired locally to a cloud device. A user manually or automatically establishes a network connection between an access device and the cloud device, and uploads the streaming data stored in the access device to the cloud device through an application or a web page.

[0003] In the related art, in the process of uploading data, when a network is interrupted, the access device will stop uploading the data or cache the acquired streaming data to a temporary folder. After the network is restored, the streaming data cached in the temporary folder is manually exported and uploaded.

SUMMARY

[0004] Embodiments of the present disclosure provide a method and apparatus for uploading streaming data, an access device, and a storage medium. The technical solutions are as follows.

[0005] In an aspect, embodiments of the present disclosure provide a method for uploading streaming data. The method is applicable to an access device and includes:

[0006] dividing the streaming data into real-time data and history data based on an acquisition timestamp of the streaming data in response to interrupting a network between the access device and a cloud device;

[0007] determining an uploading rate of the history data according to a data volume of the real-time data in response to restoring the network; and

[0008] uploading the history data based on the uploading rate during uploading the real-time data.

[0009] In another aspect, embodiments of the present disclosure provide an apparatus for uploading streaming data. The apparatus is applicable to an access device and includes:

[0010] a data dividing module configured to divide the streaming data into real-time data and history data based on an acquisition timestamp of the streaming data in response to interrupting a network between the access device and a cloud device;

[0011] a rate determining module configured to determine an uploading rate of the history data according to a data volume of the real-time data in response to restoring the network; and

[0012] a data uploading module configured to upload the history data based on the uploading rate during uploading the real-time data.

[0013] In still another aspect, embodiments of the present disclosure provide an access device. The access device includes a processor and a memory storing at least one instruction, and the processor, when loading and executing the at least one instruction, is caused to perform the method for uploading streaming data as described in the above aspect.

[0014] In yet still another aspect, embodiments of the present disclosure provide a computer-readable storage medium storing at least one instruction therein, wherein the at least one instruction, when loaded and executed by a processor, causes the processor to perform the method for uploading streaming data as described in the above aspect.

[0015] In further aspect, embodiments of the present disclosure provide a computer program product or a computer program. The computer program product or the computer program includes a computer instruction which is stored in a computer-readable storage medium. A processor of an access device reads the computer instruction from the computer-readable storage medium, and the processor executes the computer instruction, so that the access device executes the method for uploading streaming data provided in the above aspect.

[0016] The beneficial effects brought by the technical solutions provided in the embodiments of the present disclosure at least include the following.

[0017] According to the solutions provided by the embodiments of the present disclosure, after the network connection between the access device and the cloud device is interrupted, the access device can acquire continuously the streaming data and divide the streaming data into the real-time data and the history data based on the acquisition timestamp when the streaming data is acquired. After the network connection between the access device and the cloud device is restored, the access device can dynamically adjust the uploading rate of the history data according to the data volume of the real-time data. Compared with a method of manually exporting the cached data after the network is restored in the related art, the access device in the embodiments of the present disclosure can adjust the uploading rate of the history data in real time according to the data volume of the real-time data, and upload the cached history data to the cloud device punctually on the premise of ensuring that the real-time data is uploaded preferentially, which improves the timeliness for uploading of the history data.

BRIEF DESCRIPTION OF THE DRAWINGS

[0018] FIG. 1 is a schematic diagram of a computer system according to some exemplary embodiments of the present disclosure;

[0019] FIG. 2 is a flowchart of a method for uploading streaming data according to some exemplary embodiments of the present disclosure;

[0020] FIG. 3 is a flowchart of another method for uploading streaming data according to some exemplary embodiments of the present disclosure;

[0021] FIG. 4 is a flowchart of still another method for uploading streaming data according to some exemplary embodiments of the present disclosure;

[0022] FIG. 5 is a schematic diagram of a real-time data queue according to some exemplary embodiments of the present disclosure;

[0023] FIG. 6 is an algorithm flowchart of a method for uploading streaming data according to some exemplary embodiments of the present disclosure;

[0024] FIG. 7 is a graph of an uploading rate of history data according to some exemplary embodiments of the present disclosure;

[0025] FIG. 8 is a structural block diagram of an apparatus for uploading streaming data according to some exemplary embodiments of the present disclosure; and

[0026] FIG. 9 is a structural block diagram of an access device according to some exemplary embodiments of the present disclosure.

DETAILED DESCRIPTION

[0027] For clearer descriptions of the objectives, technical solutions and advantages in the present disclosure, embodiments of the present disclosure are described in detail hereinafter with reference to the accompanying drawings.

[0028] The expression "a plurality of" used herein refers to two or more. The term "and/or" describes an association relationship of associated objects, indicating that there may be three types of relationships. For example, A and/or B may indicate three cases: A exists alone, A and B exist simultaneously, and B exists alone. The character "/" generally indicates that the associated objects are in an "or" relationship.

[0029] Streaming data is a set of dynamic data that is generated in real time according to a time sequence and grows infinitely over time. As the streaming data is real-time, volatile, and infinite, it is widely used in streaming big data application scenarios, such as network monitoring, sensor networks, aerospace, meteorological measurement and control, and financial services. For example, a computer device acquires temperature data detected by a temperature sensor at a

certain time frequency, the temperature data is continuously generated and flows into the computer device, and the computer device needs to be able to run lastingly and stably. Therefore, the computer device needs to acquire the temperature data and process the temperature data or upload the temperature data to a cloud device for processing at any time.

[0030] In the related art, in a process that a user uploads data via an access device, due to the interruption of a network between the access device and a cloud device, the access device will continuously acquire streaming data, and then cache the acquired streaming data in a history folder. After the network is restored, the streaming data cached in the history folder needs to be uploaded to the cloud device by the user manually.

[0031] For some access devices that support resumable uploading, in the case of network interruption, the access device continuously acquires the streaming data, and then caches the streaming data in the history folder. After the network is restored, the access device uploads the cached history data to the cloud device at a fixed uploading rate. For an access device with poor network bandwidth, other processes running on the access device also preempt the network bandwidth, thereby resulting in the lower uploading efficiency. For an access device with better network bandwidth, the mode of uploading at a fixed uploading rate cannot make full use of the network bandwidth, and also negatively affects the uploading efficiency of the streaming data.

[0032] According to the solutions provided by the embodiments of the present disclosure, when the network is interrupted, the access device continuously acquires the streaming data needing to be uploaded, and divides the acquired streaming data into history data and real-time data based on an acquisition timestamp. After the network is restored, the access device can dynamically adjust the uploading rate of the history data based on the network bandwidth and backlog state of the real-time data. Therefore, while uploading the real-time data to the cloud device, the access device can upload the cached history data to the cloud device as soon as possible, which effectively improves the uploading efficiency of the streaming data.

[0033] FIG. 1 is a schematic diagram of a computer system according to some embodiments of the present disclosure. The computer system includes an access device 110 and a cloud device 120.

[0034] An application or a web page supporting data uploading runs in the access device 110. The access device 110 is an electronic device, such as a mobile phone, a desktop computer, a tablet computer, a multimedia playing device and a laptop portable computer, or a workstation and other devices, which is not limited in this embodiment. The application is a cloud disk application, which may have data uploading and data downloading functions; and the web page is an online web page having data uploading and downloading functions. In addition, the access device 110 may also implement the data uploading function by means of a running online applet.

In this embodiment, an application installed on the access device 110 is used as an example for description, the user may establish a connection with the cloud device 120 via the access device 110 and upload the streaming data locally acquired to the cloud device 120 for processing.

[0035] The access device 110 is connected to the cloud device 120 via a wireless network or a wired network.

[0036] The cloud device 120 may be a server or a storage device, including at least one of a server, multiple servers, a cloud computing platform, and a virtualization center. The cloud device 120 provides background services for the access device 110. For example, the cloud device 120 may be a background server of the above application. In the embodiment of the present disclosure, the cloud device 120 is configured to receive a data uploading request sent by the access device 110, and save the streaming data uploaded by the access device 110. In some exemplary embodiments, the cloud device 120 may also process and analyze the uploaded streaming data. In addition, the cloud device 120 may also provide a data downloading service, and distribute corresponding data to the access device 110 based on the received data downloading request sent by the access device 110.

[0037] A person skilled in the art may know that the number of the above access devices may be more or less. For example, the number of the above access device may be only one, or dozens or hundreds, or more. The number and type of the access device are not limited in the embodiments of the present disclosure.

[0038] FIG. 2 is a flowchart of a method for uploading data according to some exemplary embodiments of the present disclosure. The embodiments are described by taking that the method is applicable to the access device 110 shown in FIG. 1 as an example. The method includes the following steps.

[0039] In step 201, the streaming data is divided into real-time data and history data based on an acquisition timestamp of the streaming data in response to interrupting a network between the access device and a cloud device.

[0040] Before the access device is not disconnected from the cloud device, the access device acquires the streaming data that needs to be uploaded in real time and upload the same to the cloud device over the network. When the access device is disconnected from the cloud device due to the bandwidth or other factors, the access device continuously acquires the streaming data that needs to be uploaded. The streaming data is data uploaded manually by a user to the cloud device or automatically uploaded by the access device. Specifically, streaming data is pictures, videos, documents, or data acquired by the access device from other devices, etc. The specific type of streaming data is not limited in the present disclosure.

[0041] When acquiring the streaming data, the access device determines the acquisition timestamp corresponding to the streaming data based on the time when the streaming data is acquired. After the network of the access device is interrupted, the access device periodically classifies all the acquired streaming data based on the acquisition timestamp, and divides the streaming data into the real-time data and the history data. For example, by setting a cycle detection time for the access device, the acquisition timestamp of the acquired streaming data is acquired every cycle detection time. For example, if the cycle detection time is set to 1 minute, when the acquisition timestamp indicates that the streaming data is data acquired 1 minute ago, it indicates that the corresponding streaming data has not been uploaded to the cloud device punctually after being acquired, and is determined as the history data. When it is detected that the acquisition timestamp of the streaming data indicates that the streaming data is data acquired within 1 minute, it indicates that the corresponding streaming data is newly acquired data, and is determined as the real-time data.

[0042] In step 202, an uploading rate of the history data is determined according to a data volume of the real-time data in response to restoring the network.

[0043] When the network connection between the access device and the cloud device is restored, since both the history data and the real-time data are cached in the access device, in order to be able to upload the history data to the cloud device punctually without negatively affecting the acquisition and uploading of the real-time data, after the network is restored, it needs to detect the data volume of the real-time data, and determine the uploading rate of the history data according to the data volume of the real-time data.

[0044] Since the network bandwidth is fixed, when the uploading rate of history data is set to be relatively high, the uploading rate of the real-time data is relatively low; and when other processes are running on the access device at the same time, these processes also preempt part of the network bandwidth, which further decreases the uploading rate of the real-time data. In addition, since the step of acquiring data by the access device continuously proceeds, when the acquired real-time data cannot be uploaded to the cloud device punctually, the data volume of the real-time data increases. In scenarios with higher requirements on the timeliness of the streaming data, the access device needs to upload the acquired streaming data to the cloud device punctually. In the case that the access device has cached a lot of real-time data, the uploading rate of the history data should be appropriately decreased to ensure that the real-time data is uploaded to the cloud device punctually.

[0045] In step 203, the history data is uploaded based on the uploading rate during uploading the real-time data.

[0046] After determining the uploading rate of the history data according to the data volume of the real-time data, the access device further uploads the cached history data to the cloud device based on the determined uploading rate. Moreover, while uploading the history data, the access device also determines an uploading rate of the real-time data based on the network bandwidth, and synchronously uploads the cached real-time data to the cloud device.

[0047] The mode of dynamically adjusting the uploading rate of the history data according to the data volume of the real-time data can ensure that after the network is restored, the access device can make full use of the network bandwidth to upload the cached history data to the cloud device punctually without affecting the uploading of the real-time data.

[0048] In summary, according to the method provided by the embodiments of the present disclosure, after the network connection between the access device and the cloud device is interrupted, the access device can acquire continuously the streaming data and divide the streaming data into the real-time data and the history data based on the acquisition timestamp when the streaming data is acquired. After the network connection between the access device and the cloud device is restored, the access device can dynamically adjust the uploading rate of the history data according to the data volume of the real-time data. Compared with a method of manually exporting the cached data after the network is restored in the related art, the access device in the embodiments of the present disclosure can adjust the uploading rate of the history data in real time according to the data volume of the real-time data, and upload the cached history data to the cloud device punctually on the premise of ensuring that the real-time data is uploaded preferentially, which improves the timeliness for uploading of the history data.

[0049] FIG. 3 shows a flowchart of a method for uploading streaming data according to some exemplary embodiments of the present disclosure. These embodiments are described by taking that the method is applicable to the access device 110 shown in FIG. 1 as an example. The method includes the following steps.

[0050] In step 301, after a network is interrupted, the streaming data is acquired continuously, and an acquisition timestamp is determined based on the time when the streaming data is acquired.

[0051] The implementation of this step refers to step 201 and is not repeated in this embodiment.

[0052] In step 302, the streaming data is cached into a real-time data queue.

[0053] Since the access device continuously proceeds the step of acquiring the streaming data no matter whether the network connection is interrupted or restored, the access device classifies the streaming data newly acquired by the access device as the real-time data by default, and caches the same to the real-time data queue.

[0054] In step 303, an acquisition timestamp of the real-time data in the real-time data queue is acquired in response to reaching a cycle detection time, wherein the cycle detection time is a detection time periodically determined.

[0055] After caching the newly acquired real-time data to the real-time data queue, the access device further needs to divide the real-time data periodically, and move the real-time data that has been acquired but not uploaded punctually into a history data queue. The cycle detection time is determined periodically, and after the cycle detection time is reached every time, a time point when the streaming data is classified is determined, and then the streaming data is classified. Assuming that the cycle detection time is set to 5 minutes, the acquisition timestamps of all the streaming data are detected every 5 minutes, and the streaming data is divided into the history data and the real-time data based on the acquisition timestamps.

[0056] In step 304, a time difference between a detection timestamp and the acquisition timestamp is calculated based on the detection timestamp corresponding to the cycle detection time.

[0057] When the cycle detection time is reached, the detection timestamp corresponding to a current moment is determined based on the cycle detection time, and the time difference between the detection timestamp and the acquisition timestamp of the streaming data is calculated. The time difference is a cache time when the streaming data is acquired onto the access device, but not uploaded to the cloud device.

[0058] In step 305, in response to the time difference being greater than a time difference threshold, the real-time data is determined as the history data, and moved into the history data queue.

[0059] Before calculating the time difference between the detection timestamp and the acquisition timestamp, the access device also needs to set the time difference threshold in advance, and the time difference threshold is configured to determine the history data and the real-time data. When the time difference between the acquisition timestamp and the detection timestamp is greater than the time difference threshold, it indicates that the corresponding data is not uploaded to the cloud device punctually within the time difference threshold, and is re-determined as the history data. When the time difference between the acquisition timestamp and the detection timestamp is less than the time difference threshold, it indicates that the data is newest acquired data, and determined as the real-time data.

[0060] Further, the access device waits for the next cycle detection time, and divides all real-time data again when the cycle detection time is reached again. When the cache time of the real-time data in the real-time data queue exceeds the time difference threshold, the real-time data is moved into the history data queue.

[0061] The time difference threshold may be set based on the network bandwidth, the data type of the streaming data, the data volume of the uploaded data, and the access speed at which the cloud device is accessed. For example, the time difference threshold is set to 1 minute. When the real-time data is cached in the access device for more than 1 minute, the real-time data of which the cache time is greater than the time difference threshold is reclassified as the history data, and then moved into the history data queue.

[0062] In step 306, in response to the time difference being less than the time difference threshold, the real-time data is continuously cached to the real-time data queue.

[0063] It can be seen from the above content that when the time difference between the acquisition timestamp and the detection timestamp is less than the time difference threshold, it indicates that the data is newly acquired streaming data. That is, the cache time of the streaming data in the access device does not exceed the time difference threshold, and the access device at this time still determines the streaming data as the real-time data and continuously caches the same into the real-time data queue.

[0064] In some possible embodiments, the real-time data is stored in the real-time data queue in the order of acquisition timestamps. As shown in FIG. 5, first streaming data 501 to fifth streaming data 502 in the real-time data queue respectively represent data acquired by the access device in last five times and are arranged and stored sequentially in the order of acquisition timestamps. The first streaming data 501 is data acquired most early and the corresponding first acquisition timestamp is the longest from a current time; and the fifth streaming data 502 is data acquired most recently, and the corresponding fifth acquisition timestamp is shortest from the current time. When the cycle detection time is reached every time, the access device determines a detection point 503 in the real-time data queue based on the acquisition timestamp when the streaming data is acquired. The time difference between the acquisition timestamp of the streaming data corresponding to the detection point 503 and the detection timestamp is the time difference threshold. Assuming that the detection point 503 determined by the access device is a position where the third streaming data 504 is located, when the time difference between a third acquisition timestamp corresponding to the third streaming data 504 and the current time exceeds the time difference threshold, the third streaming data 504 is determined as the history data. Since the real-time data in the real-time data queue is arranged in the order of acquisition timestamps, all the data acquired before the third streaming data 504 may be determined as the history data. That is, the first streaming data 501 to the third streaming data 504 are all moved into the history data queue, and the streaming data acquired after the third streaming data 504 is still real-time data and is not moved.

[0065] In some exemplary embodiments, the history data moved into the history data queue is also arranged and stored in a cache space in the order of acquisition timestamps.

[0066] In step 307, whether the real-time data is in a backlog state is periodically determined according to the data volume of the real-time data in response to restoring the network.

[0067] After the network is restored, the access device periodically detects the data volume of the real-time data in the real-time data queue, and then determines whether the backlog of the real-time data occurs according to the data volume of the real-time data. In order to be able to dynamically adjust the uploading rate of the history data, the backlog state of the real-time data is determined by means of setting a data volume threshold. When the data volume of the real-time data does not exceed the data volume threshold as set, it indicates that the real-time data is not in the backlog state, and the uploading rate of the history data does not negatively affect the uploading of the real-time data. That is, the real-time data acquired by the access device is uploaded to the cloud device punctually. When the access device detects that the data volume of the real-time data exceeds the data volume threshold, it indicates that a large amount of real-time data has been cached in the real-time data queue and the data backlog occurs since the real-time data cannot be uploaded to the cloud device punctually.

[0068] In step 308, the uploading rate of the history data is increased in the case that the real-time data is not in the backlog state and is decreased in the case that the real-time data is in the backlog state.

[0069] In some possible embodiments, the access device sets the data volume threshold of the real-time data to 50 MB. When it is detected that the data volume of the real-time data in the real-time data queue does not reach 50 MB, it indicates that the data volume of the real-time data cached by the access device is small, and the real-time data is not in the backlog state. At this time, the access device may increase the uploading rate of the history data, and upload the cached history data as soon as possible, and in this way, the network bandwidth is used fully.

[0070] When it is detected that the data volume of the real-time data has exceeded 50 MB, it indicates that the data volume of the real-time data cached by the access device is large, and with the uploading rate of the real-time data in a current cycle, the acquired streaming data cannot be uploaded to the cloud device punctually. At this time, the access device should decrease the uploading rate of the history data to leave more network bandwidth for uploading the real-time data.

[0071] The access device detects the backlog state of the real-time data within each cycle, and then dynamically adjusts the uploading rate of the history data based on the backlog state of the real-time data. When it is detected that the real-time data is not in the backlog state in the current cycle, the uploading rate of the history data in the current cycle is increased according to a preset

rate increase method on the basis of the uploading rate of the history data in the previous cycle. Likewise, when the access device detects that the real-time data is in the backlog state in the current cycle, the access device decreases the uploading rate of the history data in the current cycle according to a preset rate decrease method on the basis of the uploading rate of the history data in the previous cycle.

[0072] Usually, in an initial stage after the network is restored, in order to be able to upload the cached history data to the cloud device as soon as possible, the access device needs to rapidly increase the uploading rate of the history data on the premise of not negatively affecting the uploading of the real-time data. At this time, the uploading rate of the history data is in a rapid increase stage. However, since the network bandwidth is fixed, while increasing the uploading rate of the history data, it needs to decrease the uploading rate of the real-time data. In the case of a larger amount of history data, after the history data is increased to certain extent, the backlog of the real-time data may occur. At this time, the access device cannot increase the uploading rate of the history data according to a preset increase strategy, but adaptively decreases or increases the uploading rate of the history data based on the backlog state of the real-time data in each cycle. As shown in FIG. 5, this process specifically includes the following steps.

[0073] In step 308A, an i^{th} uploading rate of the history data in an i^{th} cycle is acquired by increasing an $(i-1)^{\text{th}}$ uploading rate of the history data in an $(i-1)^{\text{th}}$ cycle according to a preset growth mode in the case that the real-time data in both the i^{th} cycle and a history cycle is not in the backlog state.

[0074] After the network is restored, the access device executes to the i^{th} cycle. Before determining the i^{th} uploading rate of the history data in the i^{th} cycle, the access device needs to determine a stage to which the updating rate is in in the current cycle. That is, it needs determine whether the uploading rate of the history data in the current cycle is in the rapid increase stage or a fluctuation adjustment stage. After it is determined that the uploading rate is in the rapid increase stage, the i^{th} uploading rate in the i^{th} cycle is acquired by increasing the $(i-1)^{\text{th}}$ uploading rate of the history data in the $(i-1)^{\text{th}}$ cycle according to the preset growth mode. The access device determines whether the history data is in the rapid increase stage in the i^{th} cycle based on the backlog state of the real-time data in the i^{th} cycle and the history cycle, and the determining process specifically includes the following steps.

[0075] I. A history backlog mark of the real-time data is acquired, wherein the history backlog mark is configured to indicate whether the backlog of the real-time data occurs in the history cycle.

[0076] After the network is restored, when the access device executes to the i^{th} cycle, the access device first detects the backlog state of the real-time data in the i^{th} cycle. When the access device

detects that the real-time data in the current cycle is not in the backlog state, the access device needs to determine whether the backlog of the real-time data occurs in the history cycle.

[0077] In some possible embodiments, the access device records the backlog state of the real-time data in all history cycles, and determines that the uploading rate of the history data is still in the fast increase stage in the i^{th} cycle when the access device finds that the backlog of the real-time data never occurs in the history cycle.

[0078] In other possible embodiments, the access device sets the history backlog mark during the process of uploading the history data, wherein the history backlog mark is configured to indicate whether the backlog of the real-time data occurs in the history cycle. When the history backlog mark is set to 0, it indicates that the backlog of the real-time data never occurs in the history cycle. When the history backlog mark is set to 1, it indicates that the backlog of the real-time data occurs in the history cycle. After the network is restored, the access device first sets the history backlog mark to 0, then periodically detects the backlog state of the real-time data, and sets the history backlog mark to 1 when detecting that the backlog of the real-time data occurs for the first time. In each execution cycle, the access device uses the history backlog mark as a determination basis. By means of setting the history backlog mark, the access device is prevented from storing and searching for the backlog state of the real-time data in all history cycles, so that the response speed of the access device can be increased.

[0079] II. In the case that the real-time data is not in the backlog state in the i^{th} cycle and the history backlog mark indicates that the backlog state does not occur, the i^{th} uploading rate of the history data is determined based on the $(i-1)^{\text{th}}$ uploading rate of the history data and the preset growth mode.

[0080] When the access device determines that the real-time data is not in the backlog state in the i^{th} cycle, and finds that the history backlog mark is 0, it indicates that the history data has always been in the increase stage. The access device determines the i^{th} uploading rate in the i^{th} cycle according to the preset growth mode based on the $(i-1)^{\text{th}}$ uploading rate of the history data in the $(i-1)^{\text{th}}$ cycle. The preset growth mode may be an exponential growth mode, a logarithmic growth mode, or a linear growth mode.

[0081] In some possible embodiments, the preset growth mode is an exponential-function-based growth. After the network connection is restored and the access device initializes the uploading rate of the history data, the uploading rates in different cycles are determined according to an exponential function. For example, an exponential function relationship of the uploading rate is set as $v_i = a \cdot b^{i-1}$, where i is a positive integer, and denotes the number of cycles currently executed, a is an initial uploading rate in a first cycle after the network is restored, and b is a common ratio of exponential functions. When $a = 3$ and $b = 2$, the uploading rate is expressed as

$v_i = 3 \cdot 2^{i-1}$. When $i = 1$, the initial uploading rate in the first cycle is $v_1 = 3$; and when $i = 2$, the second uploading rate in the second cycle is $v_2 = 6$, and in this way, the uploading rate in the corresponding cycle thereafter is calculated. After determining the number of currently executed cycles, or determining the uploading rate and the common ratio in the previous cycle, the access device then determines the uploading rate of the history data in the current cycle.

[0082] In addition, the preset growth mode may also be linear-function-based growth. For example, it is set $v_i = k \cdot i + v_0$, where k denotes an increase coefficient, v_0 denotes an initial uploading rate, and a cycle is executed every time, the uploading rate of the history data is numerically increased by k compared to that in the previous cycle.

[0083] It should be noted that the unit of the uploading rate is not specified in the above example. In practical application, the access device sets the unit of uploaded data based on the specific type of the uploaded data, such as KB/s, MB/s, and MB/min. In addition, a preset function relationship is also determined based on the data type or set by the user. For example, a linear function relationship is used to adjust the uploading rate of the history data. The unit of the uploading rate and the preset function relationship are not limited in this embodiment.

[0084] In the above mode, i is a positive integer greater than 1. Because the initial uploading rate is the default uploading rate of the history data in the first cycle after the network is restored, the real-time data at this time is not in the backlog state and the initial uploading rate is a relatively small value, the uploading rate is progressively increased rapidly in various subsequent cycles according to the preset growth mode.

[0085] In the above content, the process of acquiring the streaming data by the access device and the process of dividing the streaming data into the real-time data and the history data by the access device are also performed periodically based on the cycle detection time. When the network is restored and the access device detects that the real-time data is in the backlog state, the access device may upload the real-time data in the real-time data queue to the cloud device first, and then initialize the history data when the real-time data is no longer in the backlog state. That is, the initial uploading rate of the history data is set, and then the uploading rate of the history data is progressively increased periodically according to the preset growth mode.

[0086] It can be seen from the above content that the premise that the access device calculates the uploading rate according to the preset growth mode is that after the network is restored, the backlog of the real-time data never occurs in the history cycle, and thus the uploading rate of the history data is progressively increased periodically according to the preset growth mode. After the access device executes to certain cycle, due to the limitation of network bandwidth, the uploading rate of the real-time data decreases, which causes the backlog of the real-time data. In various subsequent cycles, after determining that the backlog of the real-time data occurs, the

access device needs to decrease the uploading rate of the history data according to the corresponding adjustment strategy. Therefore, step 308 further includes the following steps.

[0087] In step 308B, in the case that the real-time data in a j^{th} cycle is in the backlog state, after a $(j-1)^{\text{th}}$ uploading rate of the history data in a $(j-1)^{\text{th}}$ cycle is decreased, the j^{th} uploading rate of the history data in the j^{th} cycle is acquired.

[0088] When the access device executes to the j^{th} cycle, it is detected that the real-time data is in the backlog state at this time, which indicates that the uploading rate of the history data has entered the fluctuation adjustment stage from the rapid increase stage at this time, and the real-time data in the current cycle is in the backlog state. Therefore, the access device needs to decrease the uploading rate on the basis of the $(j-1)^{\text{th}}$ uploading rate in the $(j-1)^{\text{th}}$ cycle, and then determines the j^{th} uploading rate in the j^{th} cycle. In order to be able to accurately control the uploading rate of the history data, the access device determines the j^{th} uploading rate in the j^{th} cycle based on the uploading rates of the history data in the previous two cycles and the backlog state of the real-time data in the current and previous cycles. Specifically, the following steps are included.

[0089] I. The backlog state of the real-time data in the $(j-1)^{\text{th}}$ cycle is acquired, and an uploading rate difference between the $(j-1)^{\text{th}}$ uploading rate of the history data in the $(j-1)^{\text{th}}$ cycle and a $(j-2)^{\text{th}}$ uploading rate of the history data in a $(j-2)^{\text{th}}$ cycle is calculated.

[0090] When the access device executes to the j^{th} cycle, it needs to detect the backlog state of the real-time data in the j^{th} cycle. In addition, the access device also needs to acquire the backlog state of the real-time data in the $(j-1)^{\text{th}}$ cycle, and assuming that the $(j-1)^{\text{th}}$ uploading rate of the history data in the $(j-1)^{\text{th}}$ cycle is p and the $(j-2)^{\text{th}}$ uploading rate of the history data in the $(j-2)^{\text{th}}$ cycle is q , the difference m of the uploading rates in two consecutive cycles is calculated, wherein $m=|p-q|$, m , p and q being non-negative real numbers.

[0091] II. A j^{th} decrease rate of the history data in the j^{th} cycle is determined.

[0092] After calculating the uploading rate difference, the access device also needs to determine an adjustment coefficient based on the backlog state of the real-time data in the j^{th} cycle and the $(j-1)^{\text{th}}$ cycle. The adjustment coefficient is configured to determine the adjusting rate of the history data in various cycles, and different backlog states correspond to different adjustment coefficients. This process specifically includes the following steps.

[0093] 1. In the case that the real-time data in the $(j-1)^{\text{th}}$ cycle is in the backlog state and the real-time data in the j^{th} cycle is in the backlog state, a first adjustment coefficient of the history data in the j^{th} cycle is determined, and the j^{th} decrease rate of the history data in the j^{th} cycle is determined based on the product of the uploading rate difference and the first adjustment coefficient.

[0094] When the access device detects that the real-time data in the j^{th} cycle is in the backlog state and the real-time data in the $(j-1)^{\text{th}}$ cycle is likewise in the backlog state, it indicates that the real-time data is in the backlog state in two consecutive cycles. At this time, the access device should decrease the uploading rate of the history data on the basis of the $(j-1)^{\text{th}}$ uploading rate, determine the first adjustment coefficient k_1 of the history data in the j^{th} cycle, and then use the product of an uploading rate difference m between the $(j-1)^{\text{th}}$ uploading rate and a $(j-2)^{\text{th}}$ uploading rate and the first adjustment coefficient k_1 as the j^{th} decrease rate in the j^{th} cycle.

[0095] 2. In the case that the real-time data in the $(j-1)^{\text{th}}$ cycle is not in the backlog state and the real-time data in the j^{th} cycle is in the backlog state, a second adjustment coefficient of the history data in the j^{th} cycle is determined, and the j^{th} decrease rate of the history data in the j^{th} cycle is determined based on the product of the uploading rate difference and the second adjustment coefficient.

[0096] When the access device detects that the real-time data in the j^{th} cycle is in the backlog state and the real-time data in the $(j-1)^{\text{th}}$ cycle is not in the backlog state, it indicates that the backlog of the real-time data occurs for the first time. Or after the uploading rate enters the fluctuation adjustment stage, the real-time data changes from a state of no backlog to the backlog state. At this time, the access device should decrease the uploading rate of the history data on the basis of the $(j-1)^{\text{th}}$ uploading rate, determine the second adjustment coefficient k_2 of the history data in the j^{th} cycle, and then use the product of an uploading rate difference between the $(j-1)^{\text{th}}$ uploading rate and a $(j-2)^{\text{th}}$ uploading rate and the second adjustment coefficient k_2 as the j^{th} decrease rate in the j^{th} cycle.

[0097] Based on the above two situations, the j^{th} decrease rate c_j of the history data in the j^{th} cycle is determined, which is specifically expressed as follows:

[0098] $c_j = k_n * |p - q|$,

[0099] where k_n denotes an n^{th} adjustment coefficient, and n is 1 or 2. Usually, the real-time data in the j^{th} cycle and the $(j-1)^{\text{th}}$ cycle are in the backlog state. When this situation is compared with the situation that the real-time data in the $(j-1)^{\text{th}}$ cycle is not in the backlog state, but the real-time data in the j^{th} cycle is in the backlog state, the j^{th} uploading rate is decreased more on the basis of the $(j-1)^{\text{th}}$ uploading rate. Therefore, k_1 is a positive number not less than k_2 .

[00100] It should be noted that since in the process of step 308A, the access device needs to acquire the history backlog mark to determine whether the backlog of the real-time data occurs in the history cycle, in this step, when detecting that the backlog of the real-time data occurs for the first time, the access device should set the history backlog mark from 0 to 1 to ensure that the adjustment coefficient is determined based on the history backlog mark in each subsequent cycle.

[00101] III. A difference between the $(j-1)^{\text{th}}$ uploading rate of and the j^{th} decrease rate of the history data is determined as the j^{th} uploading rate of the history data in the j^{th} cycle.

[00102] After the access device determines a j^{th} adjusting rate in the j^{th} cycle, the access device should determine, on the basis of the $(j-1)^{\text{th}}$ uploading rate, the difference between the $(j-1)^{\text{th}}$ uploading rate and the j^{th} decrease rate as the j^{th} uploading rate of the history data in the j^{th} cycle. The j^{th} uploading rate is expressed as follows.

[00103] $v_j = p - k_n * |p - q|$,

[00104] where v_j denotes the j^{th} uploading rate of the history data in the j^{th} cycle, k_n denotes an n^{th} adjustment coefficient, and p is the $(j-1)^{\text{th}}$ uploading rate of the history data in the $(j-1)^{\text{th}}$ cycle and q is the $(j-2)^{\text{th}}$ uploading rate of the history data in the $(j-2)^{\text{th}}$ cycle.

[00105] It should be noted that in the process that the access device progressively increases the uploading rate according to the preset growth mode, the real-time data in the real-time data queue does not have the backlog state. When the j^{th} cycle indicates that the backlog of the real-time data occurs for the first time, both the $(j-1)^{\text{th}}$ uploading rate p and the $(j-2)^{\text{th}}$ uploading rate q are determined by means of the preset growth mode.

[00106] It can be seen from step 308A that the history backlog mark is configured to determine whether the backlog of the real-time data occurs in the history cycle. When the history backlog mark is 1, it indicates that the backlog of the real-time data occurs in the history cycle. After detecting that the real-time data is not in the backlog state in the j^{th} cycle, the access device also needs to increase the j^{th} uploading rate of the real-time data in the j^{th} cycle based on the backlog state of the real-time data in the $(j-1)^{\text{th}}$ cycle. Therefore, step 308 also includes the following steps.

[00107] In step 308C, in the case that the history backlog mark indicates that the backlog state has occurred, after the $(j-1)^{\text{th}}$ uploading rate of the history data in the $(j-1)^{\text{th}}$ cycle is increased, the j^{th} uploading rate of the history data in the j^{th} cycle is acquired.

[00108] When the history backlog mark acquired by the access device indicates that the backlog of the real-time data occurs in the history cycle, it indicates that the uploading rate of the history data has entered the fluctuation adjustment stage from the rapid increase stage, and the real-time data is not in the backlog state in the j^{th} cycle. At this time, the access device should increase the uploading rate on the basis of the $(j-1)^{\text{th}}$ uploading rate in the $(j-1)^{\text{th}}$ cycle, and then determine the j^{th} uploading rate in the j^{th} cycle. Therefore, the following steps are used to determine the adjustment coefficient and a j^{th} increase rate of the history data in the j^{th} cycle. Specifically the following steps are included.

[00109] I. The backlog state of the real-time data in the $(j-1)^{\text{th}}$ cycle is acquired, and an uploading rate difference between the $(j-1)^{\text{th}}$ uploading rate of the history data in the $(j-1)^{\text{th}}$ cycle and the $(j-2)^{\text{th}}$ uploading rate of the history data in the $(j-2)^{\text{th}}$ cycle is calculated.

[00110] This step refers to step 308B, which is not repeated in this embodiment.

[00111] II. The j^{th} increase rate of the history data in the j^{th} cycle is determined.

[00112] 1. In the case that the real-time data in the $(j-1)^{\text{th}}$ cycle is in the backlog state and the real-time data in the j^{th} cycle is not in the backlog state, a third adjustment coefficient of the history data in the j^{th} cycle is determined, and the j^{th} increase rate of the history data in the j^{th} cycle is determined based on the product of the uploading rate difference and the third adjustment coefficient.

[00113] In the case that the access device determines that the backlog of the real-time data occurs in the history cycle, but the real-time data in the $(j-1)^{\text{th}}$ cycle is in the backlog state and the real-time data in the j^{th} cycle is not in the backlog state, it indicates that the real-time data changes from the backlog state to the state of no backlog after the $(j-1)^{\text{th}}$ uploading rate of the history data is decreased in the $(j-1)^{\text{th}}$ cycle. The access device correspondingly determines the third adjustment coefficient k_3 of the history data in the j^{th} cycle, and then uses the product of the uploading rate difference m between the $(j-1)^{\text{th}}$ uploading rate and the $(j-2)^{\text{th}}$ uploading rate and the third adjustment coefficient k_3 as the j^{th} increase rate in the j^{th} cycle.

[00114] 2. In the case that the real-time data in both the $(j-1)^{\text{th}}$ cycle and the j^{th} cycle is not in the backlog state, a fourth adjustment coefficient of the history data in the j^{th} cycle is determined, and the j^{th} increase rate of the history data in the j^{th} cycle is determined based on the product of the uploading rate difference and the fourth adjustment coefficient.

[00115] Likewise, in the case that the access device determines that the backlog of the real-time data occurs in the history cycle, the real-time data in both the $(j-1)^{\text{th}}$ cycle and the j^{th} cycle is not in the backlog state, it indicates that the uploading rate of the history data is further increased. At this time, the access device may increase the uploading rate of the history data on the basis of the $(j-1)^{\text{th}}$ uploading rate, determine the fourth adjustment coefficient k_4 of the history data in the j^{th} cycle, and then use the product of an uploading rate difference m between the $(j-1)^{\text{th}}$ uploading rate and the $(j-2)^{\text{th}}$ uploading rate and the fourth adjustment coefficient k_4 as the j^{th} increase rate in the j^{th} cycle.

[00116] Based on the above two situations, the j^{th} increase rate c_j of the history data in the j^{th} cycle is determined, which is specifically expressed as follows:

[00117] $c_j = k_r * |p - q|$,

[00118] where k_r denotes an r^{th} adjustment coefficient, and r is 3 or 4. Usually, the real-time data in the j^{th} cycle and the $(j-1)^{\text{th}}$ cycle is not in the backlog state. When this situation is

compared with the situation that the real-time data in the $(j-1)^{\text{th}}$ cycle is in the backlog state, but the real-time data in the j^{th} cycle is not in the backlog state, the j^{th} uploading rate is increased more on the basis of the $(j-1)^{\text{th}}$ uploading rate. Therefore, k_3 is a positive number not less than k_4 .

[00119] III. The sum of the $(j-1)^{\text{th}}$ uploading rate and the j^{th} increase rate of the history data is determined as the j^{th} uploading rate of the history data in the j^{th} cycle.

[00120] After determining the j^{th} adjusting rate in the j^{th} cycle, the access device should determine, on the basis of the $(j-1)^{\text{th}}$ uploading rate, the sum of the $(j-1)^{\text{th}}$ uploading rate and the j^{th} increase rate as the j^{th} uploading rate of the history data in the j^{th} cycle. The j^{th} uploading rate may be expressed as follows:

[00121]
$$v_j = p + k_r * |p - q|,$$

[00122] where v_j denotes the j^{th} uploading rate of the history data in the j^{th} cycle, k_r denotes the r^{th} adjustment coefficient, and p is the $(j-1)^{\text{th}}$ uploading rate of the history data in the $(j-1)^{\text{th}}$ and q is the $(j-2)^{\text{th}}$ uploading rate of the history data in the $(j-2)^{\text{th}}$ cycles, respectively.

[00123] In step 309, the history data is uploaded based on the uploading rate during uploading the real-time data.

[00124] After determining the uploading rate of the history data in the corresponding cycle, the access device uploads the history data as soon as possible while ensuring that the real-time data is transmitted preferentially.

[00125] In some exemplary embodiments, during uploading by the access device, there may be a larger amount of history data cached in the history data queue. In order to ensure the timeliness of the data as much as possible, the history data in the history data queue is uploaded to the cloud device in the order of acquisition timestamps to ensure that the streaming data acquired most early is uploaded so as not to affect the timeliness of the data.

[00126] In summary, in the embodiments of the present disclosure, when the network is interrupted, the access device continuously acquires the streaming data, and caches the acquired streaming data into the real-time data queue. In a subsequent process, the access device classifies the real-time data in the real-time data queue in real time based on the acquisition timestamps, and then re-determines the real-time data whose time difference exceeds the time difference threshold as the history data, which is convenient for subsequent uploading of the history data.

[00127] When the network is restored, the access device initializes the uploading rate of the history data; and determines the uploading rate of the history data in the current cycle on the basis of the uploading rate in the previous cycle according to the preset growth mode when determining that the real-time data does not have the backlog state in the history cycle.

[00128] Furthermore, after the backlog of the real-time data occurs the backlog state for the first time, the uploading rate of the history data changes from the rapid increase stage to the

fluctuation adjustment stage. In order to accurately control the uploading rate, the access device determines the uploading rate of the history data in the current cycle based on the uploading rates of the real-time data in the previous two cycles by detecting the backlog state of the real-time data in the current cycle and the previous cycle.

[00129] Further, after determining the uploading rate of the history data in the current period, the access device uploads the history data synchronously at the calculated uploading rate during the uploading the real-time data until all the history data is uploaded to the cloud device. Compared with the solution in the prior art that the history data needs to be manually exported and uploaded after the network is restored, this solution has the advantages that there is no need to manually export the history data, and the access device can make full use of the network bandwidth, dynamically adjust the uploading rate of the history data according to the data volume of the real-time data, and upload the history data to the cloud device punctually while ensuring that the real-time data is transmitted preferentially, which improves the uploading efficiency of the streaming data.

[00130] FIG. 6 shows an algorithm flowchart of a method for uploading streaming data according to some exemplary embodiments of the present disclosure. The embodiment is described by taking that the method is applicable to the access device 110 shown in FIG. 1 as an example. The method includes the following steps.

[00131] In step 601, after the network is interrupted, the streaming data is acquired, and an acquisition timestamp is determined based on the time when the streaming data is acquired.

[00132] In step 602, the streaming data is cached to a real-time data queue.

[00133] In step 603, in response to reaching a cycle detection time, an acquisition timestamp of real-time data in the real-time data queue is acquired.

[00134] In step 604, a time difference between a detection timestamp and the acquisition timestamp is calculated based on the detection timestamp corresponding to the cycle detection time.

[00135] In step 605, whether the time difference between the detection timestamp and the acquisition timestamp is greater than a time difference threshold is determined.

[00136] In step 606, in response to the time difference being greater than the time difference threshold, the real-time data is determined as history data, and moved into a history data queue.

[00137] In step 607, in response to the time difference being less than the time difference threshold, the real-time data is continuously cached to the real-time data queue.

[00138] In step 608, after the network is restored, whether the real-time data in an r^{th} cycle is in a backlog state is detected.

[00139] When the real-time data in the r^{th} cycle is not in the backlog state, step 609 is executed, and when the real-time data in the r^{th} cycle is in the backlog state, step 615 is executed.

[00140] In step 609, whether the backlog of the real-time data occurs in a history cycle is determined.

[00141] When the real-time data in the r^{th} cycle is not in the backlog state, whether the backlog of the real-time data occurs in the history cycle is determined by acquiring a history backlog mark. When the history backlog mark is 1, step 610 is executed, and when the history backlog mark is 0, step 611 is executed.

[00142] In step 610, an r^{th} uploading rate of the history data is determined based on a $(r-1)^{\text{th}}$ uploading rate of the history data and a preset growth mode.

[00143] In step 611, whether the backlog of the real-time data in a $(r-1)^{\text{th}}$ cycle occurs is determined.

[00144] When the real-time data in the r^{th} cycle is not in the backlog state and the backlog of the real-time data occurs in the history cycle, the backlog state of the real-time data in the $(r-1)^{\text{th}}$ cycle is further acquired, and whether the real-time data in the $(r-1)^{\text{th}}$ cycle is in the backlog state is determined. When the real-time data in the $(r-1)^{\text{th}}$ cycle is in the backlog state, step 612 is executed; and when the real-time data in the $(r-1)^{\text{th}}$ cycle is not in the backlog state, step 613 is executed.

[00145] In step 612, an uploading rate difference between a $(r-1)^{\text{th}}$ uploading rate and a $(r-2)^{\text{th}}$ uploading rate is calculated, and an r^{th} increase rate of the history data in an r^{th} cycle is determined based on the product of the uploading rate difference and a third adjustment coefficient.

[00146] In step 613, the uploading rate difference between the $(r-1)^{\text{th}}$ uploading rate and the $(r-2)^{\text{th}}$ uploading rate is calculated, and the r^{th} increase rate of the history data in the r^{th} cycle is determined based on the product of the uploading rate difference and a fourth adjustment coefficient.

[00147] In step 614, the sum of the $(r-1)^{\text{th}}$ uploading rate and the r^{th} increase rate of the history data is determined as the r^{th} uploading rate of the history data in the r^{th} cycle.

[00148] Step 619 is executed after the r^{th} uploading rate of the history data in the r^{th} cycle is determined.

[00149] In step 615, whether the real-time data in the $(r-1)^{\text{th}}$ cycle is in the backlog state is determined.

[00150] When the real-time data in the r^{th} cycle is in the backlog state, the backlog state of the real-time data in the $(r-1)^{\text{th}}$ cycle is acquired, and whether the real-time data in the $(r-1)^{\text{th}}$ cycle is in the backlog state is determined. When the real-time data in the $(r-1)^{\text{th}}$ cycle is in the

backlog state, step 616 is executed; and when the real-time data in the $(r-1)^{\text{th}}$ cycle is not in the backlog state, step 617 is executed.

[00151] In step 616, the uploading rate difference between the $(r-1)^{\text{th}}$ uploading rate and the $(r-2)^{\text{th}}$ uploading rate is calculated, and an r^{th} decrease rate of the history data in the r^{th} cycle is determined based on the product of the uploading rate difference and a first adjustment coefficient.

[00152] In step 617, the uploading rate difference between the $(r-1)^{\text{th}}$ uploading rate and the $(r-2)^{\text{th}}$ uploading rate is calculated, and the r^{th} decrease rate of the history data in the r^{th} cycle is determined based on the product of the uploading rate difference and a second adjustment coefficient.

[00153] In step 618, a difference between the $(r-1)^{\text{th}}$ uploading rate of and the r^{th} decrease rate of the history data is determined as the r^{th} uploading rate of the history data in the r^{th} cycle.

[00154] In step 619, the history data is uploaded based on the uploading rate during uploading the real-time data.

[00155] A specific process of determining the uploading rate of the history data in each cycle in the above embodiments is described below with reference to specific examples. The initial uploading rate of the history data after the network is restored is set as $v_1 = 2$, the preset growth mode is set as $v_i = 2 * 2^{i-1}$, T represents the number of cycles, and S_i represents the backlog state of the real-time data in the corresponding cycle. When $S_i = 1$, it indicates that the real-time data is in the backlog state, and when $S_i = 0$, it indicates that the real-time data is not in the backlog state.

[00156] As shown in table 1, when $T = 1$ and $S_1 = 0$, the initial uploading rate $v_1 = 2$; when $T = 2$ and $S_2 = 0$, the uploading rate $v_2 = 4$ of the history data is acquired by calculation according to the preset growth mode; and by such analogy, when $T = 6$ and $S_6 = 0$, the uploading rate $v_6 = 64$. During this process, the data volume of the real-time data never exceeds a data volume threshold.

[00157] When $T = 7$ and $S_7 = 1$, it indicates that the backlog of the real-time data occurs for the first time. At this time, the uploading rate is no longer determined according to the preset growth mode, but needs to be calculated based on the uploading rate difference of the uploading rates in the previous two cycles and the backlog state of the real-time data in the current and previous cycles. As shown in table 1, when $T = 5$, $v_5 = 32$; and when $T = 6$, $v_6 = 64$ and $S_6 = 0$. It can be determined therefrom that the uploading rate difference $m = |v_6 - v_5| = 32$; the first adjustment coefficient is set as $k_1 = 1.5$, the second adjustment coefficient is set as $k_2 = 1$, the third adjustment coefficient is set as $k_3 = 0.5$, and the fourth adjustment coefficient is set as $k_4 =$

1; and thus it may be determined that when $T = 7$, $v_7 = v_6 - k_2 * |v_6 - v_5| = 32$; when $T = 8$, $S_8 = 0$, $v_8 = v_7 + k_3 * |v_7 - v_6| = 48$; when $T = 9$, $v_9 = 32$; when $T = 10$, $v_{10} = 8$; and when $T = 11$, $v_{10} = 21$.

[00158] It should be noted that the first adjustment coefficient, the second adjustment coefficient, the third adjustment coefficient and the fourth adjustment coefficient in the above example are not limited to corresponding values, and their specific values may be adjusted by themselves based on the type of data or the network environment. In addition, the user may also set the numerical values in advance according to actual needs, and the numerical values of the adjustment coefficients are not limited in this embodiment.

[00159] As shown in FIG. 7, it shows a change curve of the uploading rate of the history data after the network is restored.

[00160] Table 1

Cycle/T	Uploading rate of history data / v_i	Backlog state of real-time data / S_i
1	2	0
2	4	0
3	8	0
4	16	0
5	32	0
6	64	0
7	32	1
8	48	0
9	32	1
10	8	1
11	20	0
...	...	...

[00161] In conjunction with FIG. 7 and table 1, it can be seen that before $T = 6$, the uploading rate of the history data shows a rapid increase trend, and the access device may rapidly upload the cached history data to the cloud device, and during the progressive increase process, the uploading of the real-time data is not negatively affected. When $T = 6$, the backlog of the real-time data occurs for the first time. At this time, the access device no longer determines the uploading rate by the preset growth mode, but dynamically adjusts the uploading rate in combination with the backlog state of the real-time data and the uploading rates of the history data in the previous two cycles. Thus, the uploading rate is precisely controlled

[00162] FIG. 8 shows a structural block diagram of an apparatus for uploading streaming data according to an embodiment of the present disclosure. The apparatus includes:

- [00163] a data dividing module 801 configured to divide the streaming data into real-time data and history data based on an acquisition timestamp of the streaming data in response to interrupting a network between an access device and a cloud device;
- [00164] a rate determining module 802 configured to determine an uploading rate of the history data according to a data volume of the real-time data in response to restoring the network;
- and
- [00165] a data uploading module 803 configured to upload the history data based on the uploading rate during uploading the real-time data.
- [00166] In some exemplary embodiments, the rate determining module 802 includes:
- [00167] a detecting unit configured to periodically determine whether the real-time data is in a backlog state according to the data volume of the real-time data in response to restoring the network; and
- [00168] a rate determining unit configured to increase the uploading rate of the history data in a case that the real-time data is not in the backlog state; and decrease the uploading rate of the history data in a case that the real-time data is in the backlog state.
- [00169] In some exemplary embodiments, the rate determining unit is configured to:
- [00170] acquire an i^{th} uploading rate of the history data in an i^{th} cycle by increasing an $(i-1)^{\text{th}}$ uploading rate of the history data in an $(i-1)^{\text{th}}$ cycle according to a preset growth mode in the case that the real-time data in both the i^{th} cycle and a history cycle is not in the backlog state, where i is a positive integer greater than 1; and
- [00171] acquire a j^{th} uploading rate of the history data in a j^{th} cycle by decreasing a $(j-1)^{\text{th}}$ uploading rate of the history data in a $(j-1)^{\text{th}}$ cycle in the case that the real-time data in the j^{th} cycle is in the backlog state, where j is a positive integer greater than i .
- [00172] In some exemplary embodiments, the rate determining unit is further configured to:
- [00173] acquire the backlog state of the real-time data in the $(j-1)^{\text{th}}$ cycle and calculate an uploading rate difference between the $(j-1)^{\text{th}}$ uploading rate of the history data in the $(j-1)^{\text{th}}$ cycle and a $(j-2)^{\text{th}}$ uploading rate of the history data in a $(j-2)^{\text{th}}$ cycle;
- [00174] in a case that the real-time data in the $(j-1)^{\text{th}}$ cycle is in the backlog state and the real-time data in the j^{th} cycle is in the backlog state, determine a first adjustment coefficient of the history data in the j^{th} cycle, and determine a j^{th} decrease rate corresponding to the history data in the j^{th} cycle based on the product of the uploading rate difference and the first adjustment coefficient; or
- [00175] in a case that the real-time data in the $(j-1)^{\text{th}}$ cycle is not in the backlog state and the real-time data in the j^{th} cycle is in the backlog state, determine a second adjustment

coefficient of the history data in the j^{th} cycle, and determine the j^{th} decrease rate corresponding to the history data in the j^{th} cycle based on the product of the uploading rate difference and the second adjustment coefficient; and

[00176] determine a difference between the $(j-1)^{\text{th}}$ uploading rate of and the j^{th} decrease rate of the history data as the j^{th} uploading rate of the history data in the j^{th} cycle.

[00177] In some exemplary embodiments, the rate determining unit is further configured to:

[00178] acquire a history backlog mark of the real-time data, wherein the history backlog mark is configured to indicate whether the backlog of the real-time data occurs in the history cycle; and

[00179] in a case that the real-time data in the i^{th} cycle is not in the backlog state and the history backlog mark indicates that the backlog does not occur, determine the i^{th} uploading rate of the history data based on the $(i-1)^{\text{th}}$ uploading rate of the history data and the preset growth mode.

[00180] In some exemplary embodiments, the rate determining unit is further configured to:

[00181] acquire the j^{th} uploading rate of the history data in the j^{th} cycle by increasing the $(j-1)^{\text{th}}$ uploading rate of the history data in the $(j-1)^{\text{th}}$ cycle in a case that the history backlog mark indicates that the backlog has occurred.

[00182] In some exemplary embodiments, the rate determining unit is further configured to:

[00183] acquire the backlog state of the real-time data in the $(j-1)^{\text{th}}$ cycle and calculate an uploading rate difference between the $(j-1)^{\text{th}}$ uploading rate of the history data in the $(j-1)^{\text{th}}$ cycle and a $(j-2)^{\text{th}}$ uploading rate of the history data in a $(j-2)^{\text{th}}$ cycle;

[00184] in a case that the real-time data in the $(j-1)^{\text{th}}$ cycle is in the backlog state and the real-time data in the j^{th} cycle is not in the backlog state, determine a third adjustment coefficient of the history data in the j^{th} cycle, and determine a j^{th} increase rate corresponding to the history data in the j^{th} cycle based on the product of the uploading rate difference and the third adjustment coefficient;

[00185] or in a case that the real-time data in both the $(j-1)^{\text{th}}$ cycle and the j^{th} cycle is not in the backlog state, determine a fourth adjustment coefficient of the history data in the j^{th} cycle, and determine the j^{th} increase rate corresponding to the history data in the j^{th} cycle based on the product of the uploading rate difference and the fourth adjustment coefficient; and

[00186] determine the sum of the $(j-1)^{\text{th}}$ uploading rate and the j^{th} increase rate of the history data as the j^{th} uploading rate of the history data in the j^{th} cycle.

[00187] In some exemplary embodiments, the data dividing module 801 includes:

[00188] a first determining unit configured to acquire the streaming data and determine the acquisition timestamp based on the time when the streaming data is acquired;

[00189] a second determining unit configured to cache the streaming data to a real-time data queue;

[00190] an acquiring unit configured to: in response to reaching a cycle detection time, acquire the acquisition timestamp of the real-time data in the real-time data queue, wherein the cycle detection time is a detection time periodically determined;

[00191] a calculating unit configured to calculate a time difference between a detection timestamp and the acquisition timestamp based on the detection timestamp corresponding to the cycle detection time;

[00192] a third determining unit configured to: in response to the time difference being greater than a time difference threshold, determine the real-time data as the history data, and move the same to a history data queue; and

[00193] a fourth determining unit configured to: in response to the time difference being less than the time difference threshold, continuously cache the real-time data to the real-time data queue.

[00194] FIG. 9 shows a structural schematic diagram of an access device according to some exemplary embodiments of the present disclosure. Specifically, the access device includes a central processing unit (CPU) 901, a system memory 904 including a random-access memory 902 and a read-only memory 903, and a system bus 905 connecting the system memory 904 and the central processing unit 901. The access device 900 further includes a basic input/output system (I/O system) 906 which helps information transmission between various components within a computer, and a high-capacity storage device 907 for storing an operating system 913, an application 914 and other program modules 915.

[00195] The basic input/output system 906 includes a display 908 for displaying information and an input device 909, such as a mouse and a keyboard, for inputting information by the user. Both the display 908 and the input device 909 are connected to the central processing unit 901 through an input/output controller 910 connected to the system bus 905. The basic input/output system 906 may also include the input/output controller 910 for receiving and processing input from a plurality of other devices, such as the keyboard, the mouse, or an electronic stylus. Similarly, the input/output controller 910 further provides output to the display, a printer or other types of output devices.

[00196] The high-capacity storage device 907 is connected to the central processing unit 901 through a high-capacity storage controller (not shown) connected to the system bus 905. The

high-capacity storage device 907 and a computer-readable medium associated therewith provide non-volatile storage for the access device 900. That is, the high-capacity storage device 907 may include the computer-readable medium (not shown), such as a hard disk or a driver.

[00197] Without loss of generality, the computer-readable medium may include a computer storage medium and a communication medium. The computer storage medium includes volatile and non-volatile, removable and non-removable media implemented in any method or technology for storage of information such as a computer-readable instruction, a data structure, a program module or other data. The computer storage medium includes a random-access memory (RAM), a read only memory (ROM), a flash memory or other solid-state storage technologies; a compact disc read-only memory (CD-ROM), a digital versatile disc (DVD) or other optical storage; and a tape cartridge, a magnetic tape, a disk storage or other magnetic storage devices. Of course, it will be known by a person skilled in the art that the computer storage medium is not limited to above. The above system memory 904 and the high-capacity storage device 907 may be collectively referred to as the memory.

[00198] The memory stores one or more programs. The one or more programs are configured to be executed by one or more central processing units 901 and include instructions for implementing the above method, and the central processing unit 901 executes the one or more programs to implement the method according to the above various method embodiments.

[00199] According to various embodiments of the present disclosure, the access device 900 may also be connected to a remote computer on a network through the network, such as the Internet, for operation. That is, the access device 900 may be connected to the network 912 through a network interface unit 911 connected to the system bus 905, or may be connected to other types of networks or remote computer systems (not shown) with the network interface unit 911.

[00200] The memory further includes one or more programs which are stored in the memory, and includes steps executed by the access device in the method according to the embodiments of the present disclosure.

[00201] In addition, it may be understood by a person skilled in the art that the structure of an electronic device shown in the above drawings do not constitute the limitation on the electronic device 500 and the electronic device may include more or fewer components than those shown, or combine certain components, or use different component layouts. For example, the electronic device further includes components such as a radio frequency circuit, an input unit, a sensor, a server, and a power source, which is not repeated here.

[00202] Some embodiments of the present disclosure further provide a computer-readable storage medium storing at least one instruction therein, the at least one instruction, when loaded

and executed by a processor, causes the processor to perform the method for uploading streaming data according to any one of the above embodiments.

[00203] Some embodiments of the present disclosure provide a computer program product or a computer program. The computer program product or the computer program includes a computer instruction which is stored in a computer-readable storage medium. A processor of an access device reads the computer instruction from the computer-readable storage medium, and the processor executes the computer instruction, so that the access device executes the method for uploading streaming data according to any one of the above embodiments.

[00204] A person skilled in the art shall appreciate that in one or more examples described above, the functions described in the embodiments of the present disclosure may be implemented in hardware, software, firmware, or any combination thereof. If the functions are implemented in the software, they may be stored in a computer-readable medium or transmitted as one or more instructions or codes on a computer-readable medium. The computer-readable medium includes a computer storage medium and a communication medium, wherein the communication medium includes any medium that facilitates transfer of a computer program from one place to another, and the storage medium may be any available medium that can be accessed by a general-purpose or special-purpose computer.

[00205] Described above are example embodiments of the present disclosure, and are not intended to limit the present disclosure. Any modifications, equivalent replacements, improvements and the like made within the concept and principles of the present disclosure should be included within the scope of protection of the present disclosure.

CLAIMS

What is claimed is:

1. A method for uploading streaming data, the method being applicable to an access device and comprising:

dividing the streaming data into real-time data and history data based on an acquisition timestamp of the streaming data in response to interrupting a network between the access device and a cloud device;

determining an uploading rate of the history data according to a data volume of the real-time data in response to restoring the network; and

uploading the history data based on the uploading rate during uploading the real-time data, wherein the determining the uploading rate of the history data according to the data volume of the real-time data in response to restoring the network comprises:

periodically determining whether the real-time data is in a backlog state according to the data volume of the real-time data in response to restoring the network; and

increasing the uploading rate of the history data in an i^{th} cycle in a case that the real-time data in both the i^{th} cycle and a history cycle is not in the backlog state, where i is a positive integer greater than 1.

2. The method according to claim 1, wherein the determining the uploading rate of the history data according to the data volume of the real-time data in response to restoring the network further comprises:

decreasing the uploading rate of the history data in a case that the real-time data is in the backlog state.

3. The method according to claim 2, wherein the increasing the uploading rate of the history data in the case that the real-time data is not in the backlog state comprises:

acquiring the i^{th} uploading rate of the history data in the i^{th} cycle by increasing an $(i-1)^{\text{th}}$ uploading rate of the history data in an $(i-1)^{\text{th}}$ cycle according to a preset growth mode in the case that the real-time data in both the i^{th} cycle and the history cycle is not in the backlog state, where i is a positive integer greater than 1; and

the decreasing the uploading rate of the history data in the case that the real-time data is in the backlog state comprises:

acquiring a j^{th} uploading rate of the history data in a j^{th} cycle by decreasing a $(j-1)^{\text{th}}$ uploading rate of the history data in a $(j-1)^{\text{th}}$ cycle in the case that the real-time data in the j^{th} cycle is in the backlog state, where j is a positive integer greater than 1.

4. The method according to claim 3, wherein the acquiring the j^{th} uploading rate of the history data in the j^{th} cycle by decreasing the $(j-1)^{\text{th}}$ uploading rate of the history data in the $(j-1)^{\text{th}}$ cycle comprises:

acquiring the backlog state of the real-time data in the $(j-1)^{\text{th}}$ cycle and calculating an uploading rate difference between the $(j-1)^{\text{th}}$ uploading rate of the history data in the $(j-1)^{\text{th}}$ cycle and a $(j-2)^{\text{th}}$ uploading rate of the history data in a $(j-2)^{\text{th}}$ cycle;

in a case that the real-time data in the $(j-1)^{\text{th}}$ cycle is in the backlog state and the real-time data in the j^{th} cycle is in the backlog state, determining a first adjustment coefficient of the history data in the j^{th} cycle, and determining a j^{th} decrease rate corresponding to the history data in the j^{th} cycle based on a product of the uploading rate difference and the first adjustment coefficient; or in a case that the real-time data in the $(j-1)^{\text{th}}$ cycle is not in the backlog state and the real-time data in the j^{th} cycle is in the backlog state, determining a second adjustment coefficient of the history data in the j^{th} cycle, and determining a j^{th} decrease rate corresponding to the history data in the j^{th} cycle based on a product of the uploading rate difference and the second adjustment coefficient; and

determining a difference between the $(j-1)^{\text{th}}$ uploading rate of and the j^{th} decrease rate of the history data as the j^{th} uploading rate of the history data in the j^{th} cycle.

5. The method according to claim 3, wherein the acquiring the i^{th} uploading rate of the history data in the i^{th} cycle by increasing the $(i-1)^{\text{th}}$ uploading rate of the history data in the $(i-1)^{\text{th}}$ cycle according to the preset growth mode comprises:

acquiring a history backlog mark of the real-time data, wherein the history backlog mark is configured to indicate whether the backlog of the real-time data occurs in the history cycle; and

in a case that the real-time data in the i^{th} cycle is not in the backlog state and the history backlog mark indicates that the backlog does not occur, determining the i^{th} uploading rate of the history data based on the $(i-1)^{\text{th}}$ uploading rate of the history data and the preset growth mode.

6. The method according to claim 5, further comprising:

acquiring the j^{th} uploading rate of the history data in the j^{th} cycle by increasing the $(j-1)^{\text{th}}$ uploading rate of the history data in the $(j-1)^{\text{th}}$ cycle in a case that the history backlog mark indicates that the backlog occurs.

7. The method according to claim 6, wherein the acquiring the j^{th} uploading rate of the history data in the j^{th} cycle by increasing the $(j-1)^{\text{th}}$ uploading rate of the history data in the $(j-1)^{\text{th}}$ cycle comprises:

acquiring the backlog state of the real-time data in the $(j-1)^{\text{th}}$ cycle and calculating an uploading rate difference between the $(j-1)^{\text{th}}$ uploading rate of the history data in the $(j-1)^{\text{th}}$ cycle and a $(j-2)^{\text{th}}$ uploading rate of a history data in a $(j-2)^{\text{th}}$ cycle;

in a case that the real-time data in the $(j-1)^{\text{th}}$ cycle is in the backlog state and the real-time data in the j^{th} cycle is not in the backlog state, determining a third adjustment coefficient of the history data in the j^{th} cycle, and determining a j^{th} increase rate corresponding to the history data in the j^{th} cycle based on a product of the uploading rate difference and the third adjustment coefficient; or in a case that the real-time data in both the $(j-1)^{\text{th}}$ cycle and the j^{th} cycle is not in the backlog state, determining a fourth adjustment coefficient of the history data in the j^{th} cycle, and determining a j^{th} increase rate corresponding to the history data in the j^{th} cycle based on a product of the uploading rate difference and the fourth adjustment coefficient; and

determining a sum of the $(j-1)^{\text{th}}$ uploading rate and the j^{th} increase rate of the history data as the j^{th} uploading rate of the history data in the j^{th} cycle.

8. The method according to any one of claims 1 to 7, wherein the dividing the streaming data into the real-time data and the history data based on the acquisition timestamp of the streaming data comprises:

acquiring the streaming data and determining the acquisition timestamp based on a time when the streaming data is acquired;

caching the streaming data to a real-time data queue;

acquiring the acquisition timestamp of the real-time data in the real-time data queue in response to reaching a cycle detection time, wherein the cycle detection time is a detection time periodically determined;

calculating a time difference between a detection timestamp and the acquisition timestamp based on the detection timestamp corresponding to the cycle detection time;

in response to the time difference being greater than a time difference threshold, determining the real-time data as the history data, and moving the same to a history data queue; and

in response to the time difference being less than the time difference threshold, continuously caching the real-time data to the real-time data queue.

9. An apparatus for uploading streaming data, the apparatus being applicable to an access device and comprising:

a data dividing module configured to divide the streaming data into real-time data and history data based on an acquisition timestamp of the streaming data in response to interrupting a network between the access device and a cloud device;

a rate determining module configured to determine an uploading rate of the history data according to a data volume of the real-time data in response to restoring the network;

a data uploading module configured to upload the history data based on the uploading rate during uploading the real-time data,

wherein the determining the uploading rate of the history data according to the data volume of the real-time data in response to restoring the network comprises:

periodically determining whether the real-time data is in a backlog state according to the data volume of the real-time data in response to restoring the network; and

increasing the uploading rate of the history data in an i^{th} cycle in a case that the real-time data in both the i^{th} cycle and a history cycle is not in the backlog state, where i is a positive integer greater than 1.

10. An access device, comprising a processor and a memory storing at least one instruction, wherein the processor, when loading and executing the at least one instruction, is caused to perform:

dividing the streaming data into real-time data and history data based on an acquisition timestamp of the streaming data in response to interrupting a network between the access device and a cloud device;

determining an uploading rate of the history data according to a data volume of the real-time data in response to restoring the network; and

uploading the history data based on the uploading rate during uploading the real-time data, wherein the determining the uploading rate of the history data according to the data volume of the real-time data in response to restoring the network comprises:

periodically determining whether the real-time data is in a backlog state according to the data volume of the real-time data in response to restoring the network; and

increasing the uploading rate of the history data in an i^{th} cycle in a case that the real-time data in both the i^{th} cycle and a history cycle is not in the backlog state, where i is a positive integer greater than 1.

11. A computer-readable storage medium storing at least one instruction therein, wherein the at least one instruction, when loaded and executed by a processor, causes the processor to perform:

dividing the streaming data into real-time data and history data based on an acquisition timestamp of the streaming data in response to interrupting a network between the access device and a cloud device;

determining an uploading rate of the history data according to a data volume of the real-time data in response to restoring the network; and

uploading the history data based on the uploading rate during uploading the real-time data, wherein the determining the uploading rate of the history data according to the data volume of the real-time data in response to restoring the network comprises:

periodically determining whether the real-time data is in a backlog state according to the data volume of the real-time data in response to restoring the network; and

increasing the uploading rate of the history data in an i^{th} cycle in a case that the real-time data in both the i^{th} cycle and a history cycle is not in the backlog state, where i is a positive integer greater than 1.