



(19) **United States**

(12) **Patent Application Publication**
Xie et al.

(10) **Pub. No.: US 2012/0240119 A1**
(43) **Pub. Date: Sep. 20, 2012**

(54) **METHOD AND DEVICE FOR FILE TRANSFER PROTOCOL DEADLOCK DETECTION AND SELF RECOVERY**

Publication Classification

(51) **Int. Cl.**
G06F 9/46 (2006.01)

(75) Inventors: **Xingfeng Xie**, Shenzhen (CN); **Sujian Liu**, Shenzhen (CN); **Hua Chen**, Shenzhen (CN); **Ping Zhu**, Shenzhen (CN)

(52) **U.S. Cl.** **718/100**

(73) Assignee: **ZTE CORPORATION**, Shenzhen, Guangdong (CN)

(57) **ABSTRACT**

(21) Appl. No.: **13/258,312**

A method and a device for file transfer protocol (FTP) deadlock detection and self recovery are provided by the disclosure in order to solve the sudden deadlock problem in the FTP upload operation. The method includes: if a daemon determines that a deadlock occurs in an FTP upload task by a heartbeat detection mechanism, the socket resources used by the FTP upload operation is recorded at the storage location in a socket resource cycle queue, and an FTP upload task end process is started; determining whether the socket resource cycle queue is full, if it is not full, the socket resource information occupied by the current deadlock is put into the socket resource cycle queue, otherwise, the earliest socket resources in the socket resource cycle queue are released, and the socket resource information occupied by the current deadlock is put into the socket resource cycle queue.

(22) PCT Filed: **May 25, 2010**

(86) PCT No.: **PCT/CN2010/073194**

§ 371 (c)(1),
(2), (4) Date: **May 29, 2012**

(30) **Foreign Application Priority Data**

Dec. 7, 2009 (CN) 200910254305.4

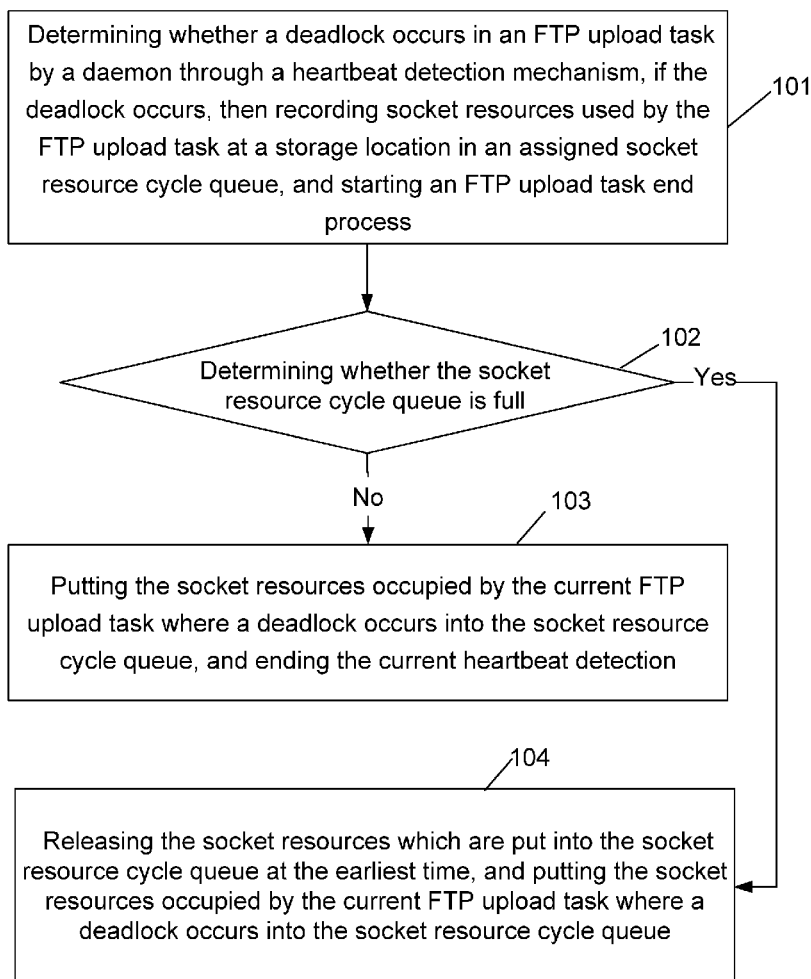


Fig. 1

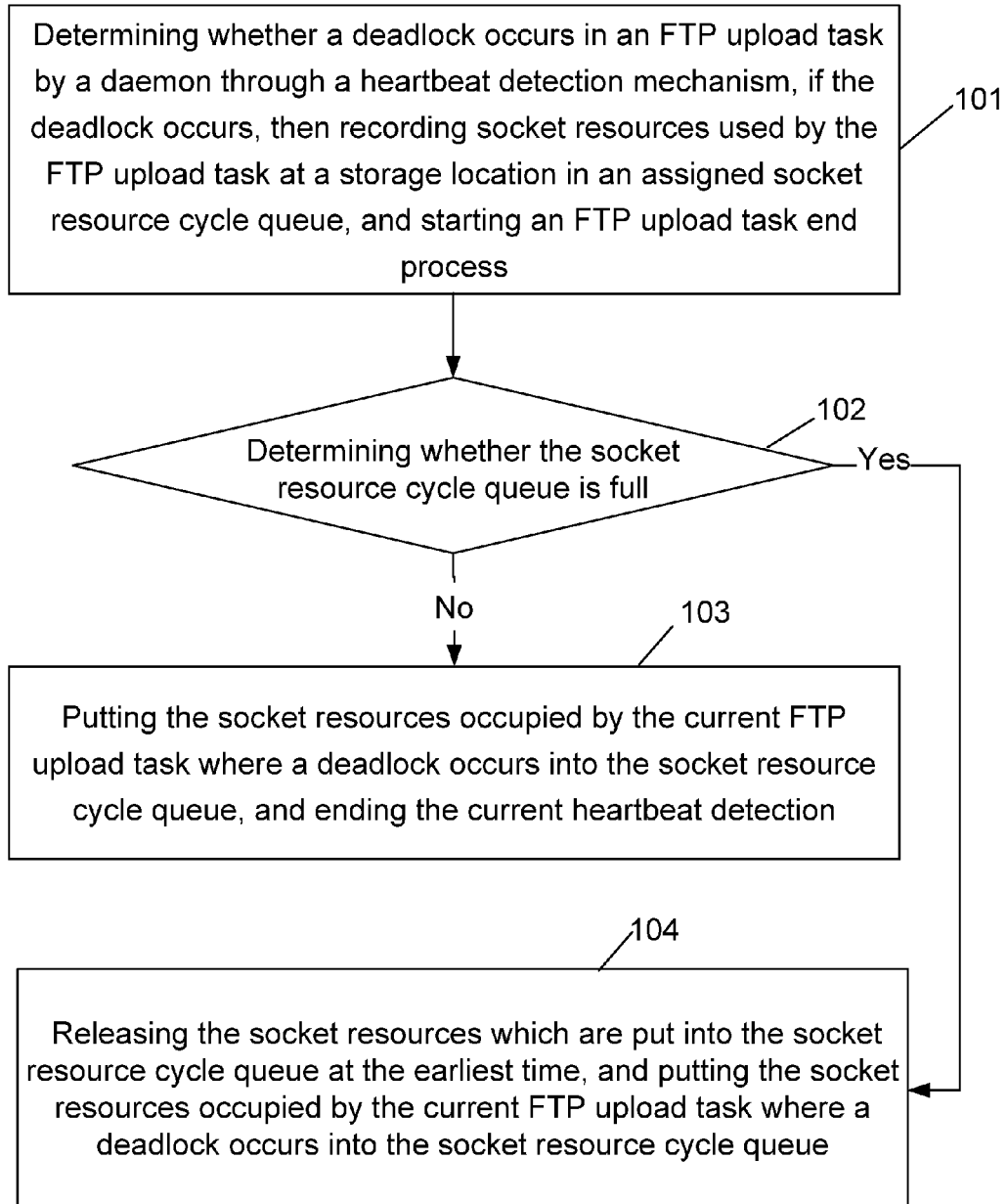


Fig. 2

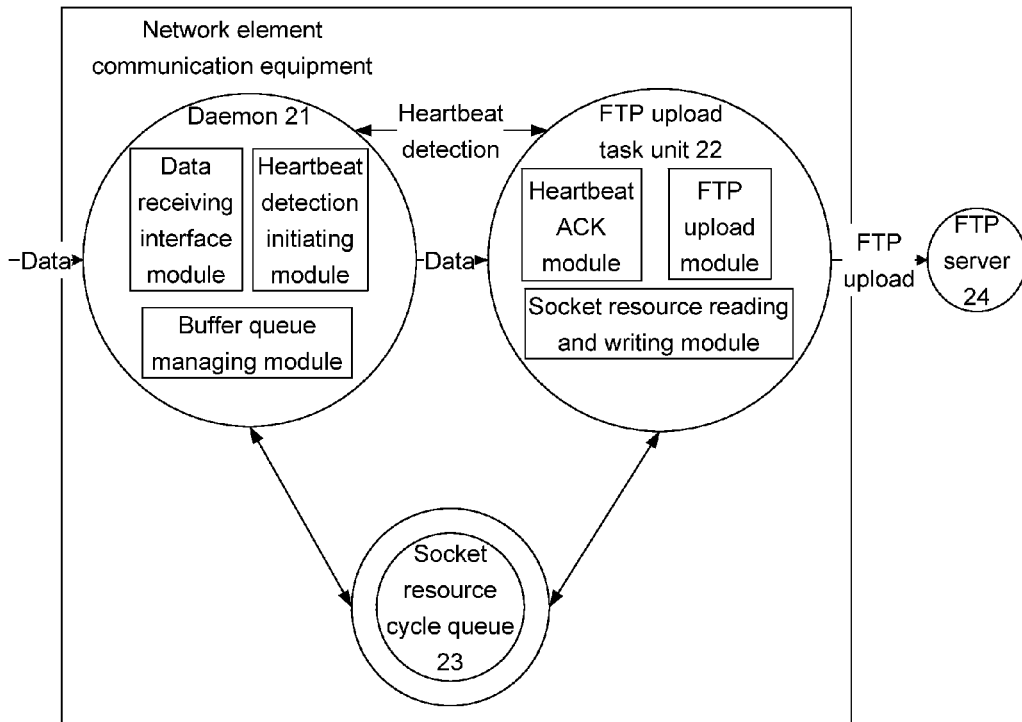


Fig. 3

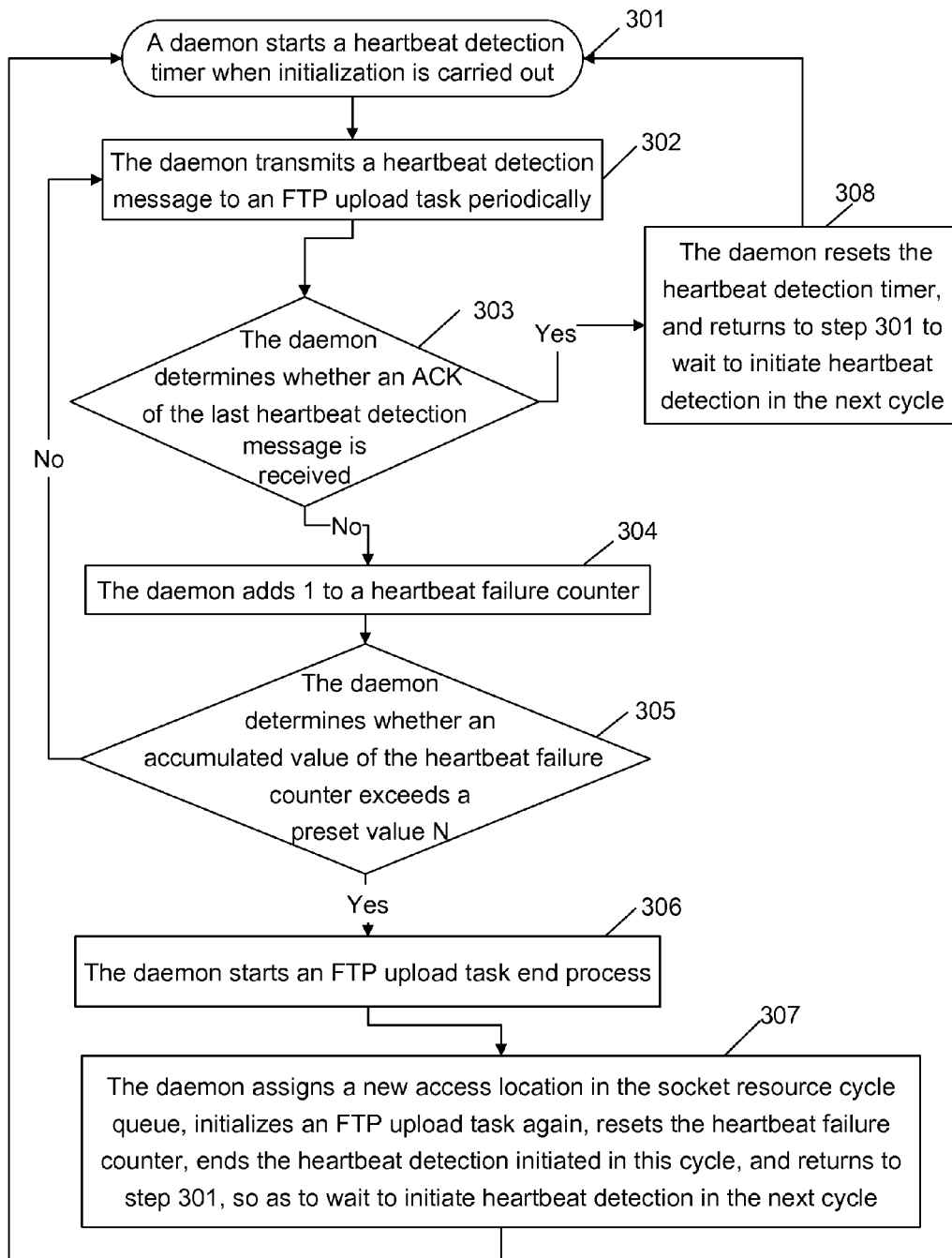
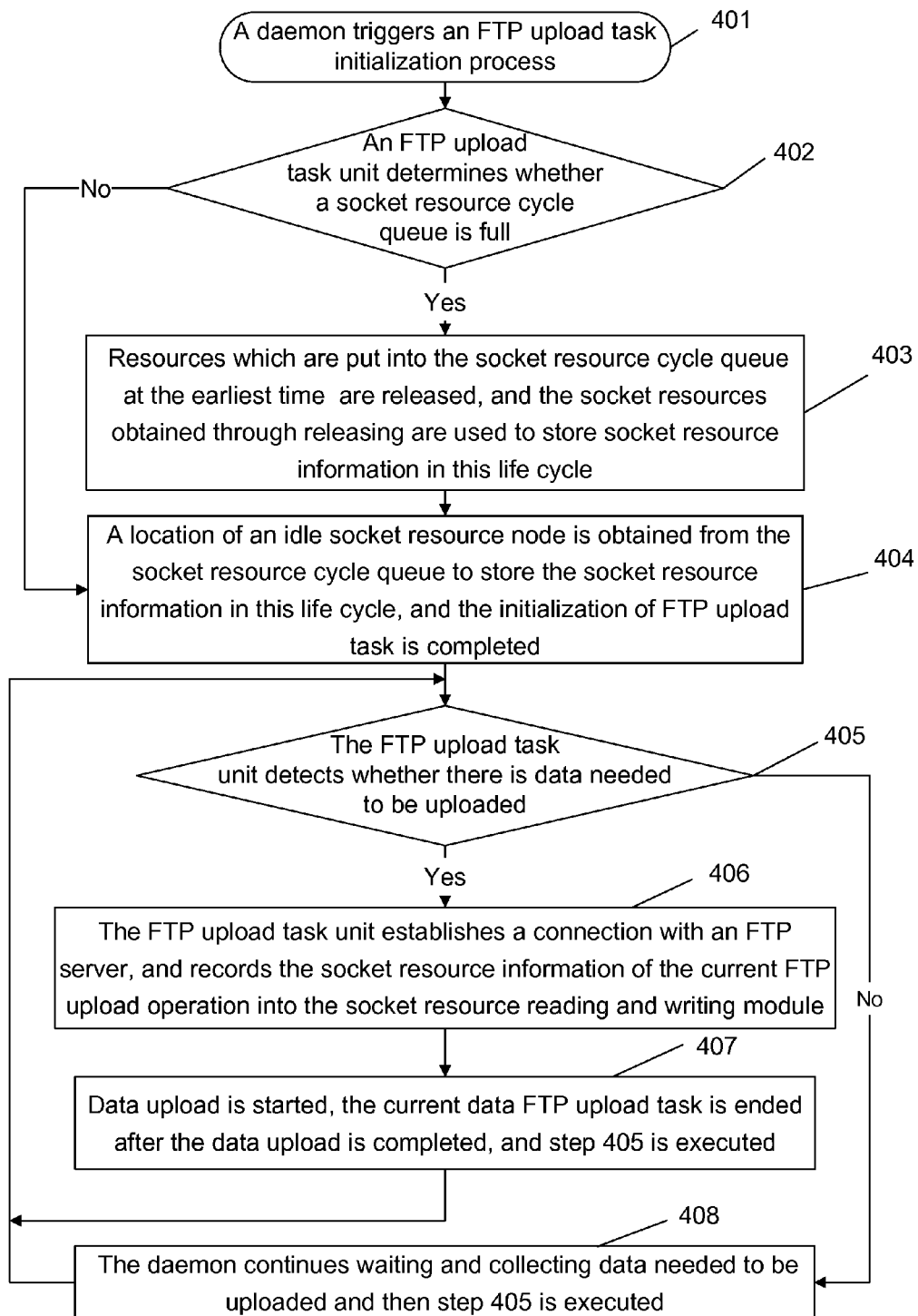


Fig. 4



**METHOD AND DEVICE FOR FILE
TRANSFER PROTOCOL DEADLOCK
DETECTION AND SELF RECOVERY**

TECHNICAL FIELD

[0001] The present disclosure relates to the File Transfer Protocol (FTP) technology in the technical field of communications, particularly to a method and a device for FTP deadlock detection and self recovery.

BACKGROUND

[0002] In mobile communication equipment, data collected from base station equipment comprise performance data of the base station, detailed information data of each call, alarm data and baseband grasping frame data and so on; the collected data have features such as large volume, real-time property, continuity and so on. The base station is connected with an Operation Maintenance Centre (OMC) through the Ethernet, and an FTP with high data throughput is the first choice for data uploading. An Ethernet network environment is complex and FTP underlying protocol processing in a vxWorks real-time operation system produced by the Wind River System Inc. is unstable, therefore, such cases as close, write, read, select, abnormal deadlock or apparent death may suddenly occur in an FTP upload operation, which are characterized in that socket resources (a control connection socket and a data connection socket) cannot be released within operation time allowed by an operation system and exit this system scheduling, therefore, process scheduling of the real-time operation system goes out of control, a single board is reset, and robustness of the base station equipment is seriously affected; in view of this, it becomes very important to perform deadlock detection and deadlock recovery to the FTP upload operation.

[0003] With regard to deadlock detection of a process, the patent Method for detecting a deadlock on Semaphore (CN200510053781) describes whether a deadlock happens to semaphore is detected by the establishment of a list of relationships between processes and semaphore and according to whether a resource wait loop determination condition exists, so as to obtain the process corresponding to the semaphore to which a deadlock happens. The disadvantage of the patent is: a deadlock can not be determined until a resource wait loop appears; for an FTP upload operation, sudden apparent death of a process, which occurs due to network environment factors, cannot be detected.

SUMMARY

[0004] The present disclosure provides a method and a device for FTP deadlock detection and self recovery, which can solve the problem of a sudden deadlock in an FTP upload operation.

[0005] A method for FTP deadlock detection and self recovery in a communication system, which is provided by an embodiment of the present disclosure, includes the following steps:

[0006] determining, by a daemon configured in a network element, whether a deadlock occurs in an FTP upload task through a heartbeat detection mechanism, if the deadlock occurs, then recording socket resources used by an FTP upload operation at a storage location in a socket resource cycle queue which is created in the network element, and starting an FTP upload task end process; and

[0007] determining whether the socket resource cycle queue is full, if it is not full, then putting socket resource information occupied by the current deadlock into the socket resource cycle queue, if it is full, then releasing earliest socket resources from the socket resource cycle queue, and putting the socket information occupied by the current deadlock into the socket resource cycle queue.

[0008] In the abovementioned method, the step of determining by the daemon whether a deadlock occurs in the FTP upload task through the heartbeat detection mechanism may include:

[0009] A1: transmitting, by the daemon, a heartbeat detection message to the FTP upload task periodically, and determining whether a response message of the heartbeat detection message is received within a certain period of time, if the response message is received, then resetting a heartbeat detection timer, and returning to step A1; otherwise, executing step A2;

[0010] A2: adding 1 to a heartbeat failure counter, and determining whether a value of the heartbeat failure counter is greater than a preset value, if the value is greater than the preset value, then determining that the deadlock occurs in the FTP upload task; otherwise, executing step A1.

[0011] In the abovementioned method, after determining that the deadlock occurs in the FTP upload task, the method may further include: resetting the heartbeat failure counter by the daemon, and returning to step A1 to initiate heartbeat detection in a next cycle, wherein a task priority to which the daemon belongs is higher than that of the FTP upload task, and the socket resources include a data connection socket, and/or a control connection socket.

[0012] Correspondingly, the present disclosure further provides a device for FTP deadlock detection and self recovery in a communication system, including: a heartbeat detection initiating module and a buffer queue managing module, wherein

[0013] the heartbeat detection initiating module is configured to determine whether a deadlock occurs in an FTP upload task through an heartbeat detection mechanism; the buffer queue managing module is configured to record socket resources used by an FTP upload operation at a storage location in an assigned socket resource cycle queue, start an FTP upload task end process, and further configured to determine whether the socket resource cycle queue is full, if it is not full, then put socket resource information occupied by the current deadlock into the socket resource cycle queue, if it is full, then release earliest socket resources from the socket resource cycle queue, and put the socket resource information occupied by the current deadlock into the socket resource cycle queue.

[0014] In the abovementioned device, the heartbeat detection initiating module may include: a transmitting module and a determining module, wherein the transmitting module is configured to transmit a heartbeat detection message to the FTP upload task periodically; the determining module is configured to determine whether a response message of the heartbeat detection message is received within a certain period of time, if the response message is received, then reset a heartbeat detection timer, and notify the transmitting module to periodically transmit a heartbeat detection message to the FTP upload task, if the response message is not received, then add 1 to a heartbeat failure counter; and the determining module is further configured to determine whether a value of

the heartbeat failure counter is greater than a preset value, if the value is greater than the preset value, then determine that the deadlock occurs in the FTP upload task, otherwise, periodically transmit a heartbeat detection message to the FTP upload task;

[0015] wherein the buffer queue managing module can be further configured to notify the determining module to reset the heartbeat failure counter after determining that the deadlock occurs in the FTP upload task; wherein a task priority to which the daemon belongs is higher than that of the FTP upload task; the socket resources include a data connection socket, and/or a control connection socket.

[0016] Correspondingly, the present disclosure further provides a system for FTP deadlock detection and self recovery, including: a daemon configured to determine that a deadlock occurs in an FTP upload task through a heartbeat detection mechanism, and then record socket resources used by an FTP upload operation at a storage location in an assigned socket resource cycle queue, and start an FTP upload task end process; and further configured to determine whether the socket resource cycle queue is full, if it is not full, then put socket resource information occupied by the current deadlock into the socket resource cycle queue, otherwise, release earliest socket resources from the socket resource cycle queue, and put the socket resource information occupied by the current deadlock into the socket resource cycle queue; an FTP upload task unit configured to return a heartbeat response in response to the heartbeat detection of the daemon, acquire data from the daemon, establish a connection with the FTP and upload the acquired data to an FTP server;

[0017] a socket resource cycle queue configured to store socket resources used by the FTP upload operation each time a deadlock occurs; and an FTP server configured to receive data uploaded by the FTP upload task.

[0018] In the abovementioned system, the daemon may include: a heartbeat detection initiating module configured to determine whether the deadlock occurs in the FTP upload task through the heartbeat detection mechanism; a buffer queue managing module configured to record the socket resources used by the FTP upload operation at the storage location in the assigned socket resource cycle queue, and start the FTP upload task end process; and further configured to determine whether the socket resource cycle queue is full, if it is not full, then put the socket resource information occupied by the current deadlock into the socket resource cycle queue, if it is full, then release the earliest socket resources from the socket resource cycle queue, and put the socket resource information occupied by the current deadlock into the socket resource cycle queue.

[0019] Compared with the prior art, embodiments of the present disclosure provide a method for detecting a deadlock by adopting a heartbeat detection mechanism, a method of releasing socket resources with delay from a socket resource cycle queue when a deadlock occurs, a self recovery method and the like. By adopting a heartbeat detection mechanism and a self recovery scheme in the embodiments of the present disclosure, problems, such as detection of a sudden deadlock in an FTP upload operation, are solved, thereby abnormal resetting of a base station single board is decreased, the operation stability of base station equipment is improved, and the impact is to system CPU, when a sudden situation appears, is decreased.

BRIEF DESCRIPTION OF THE DRAWINGS

[0020] FIG. 1 shows a flowchart of a method for FTP deadlock detection and self recovery according to an embodiment of the present disclosure;

[0021] FIG. 2 shows a structural diagram of a system for FTP deadlock detection and self recovery according to an embodiment of the present disclosure;

[0022] FIG. 3 shows a flowchart of heartbeat detection and self recovery according to an embodiment of the present disclosure; and

[0023] FIG. 4 shows a flowchart of initialization and data upload of an FTP upload task according to an embodiment of the present disclosure.

DETAILED DESCRIPTION

[0024] As shown in FIG. 1, a method for FTP deadlock detection and self recovery according to an embodiment of the present disclosure comprises the following steps:

[0025] step 101: determining by a daemon whether a deadlock occurs in an FTP upload task through a heartbeat detection mechanism, if the deadlock occurs, then recording socket resources used by the FTP upload task at a storage location in an assigned socket resource cycle queue, and starting an FTP upload task end process;

[0026] wherein, the socket resource cycle queue is a first-in first-out (FIFO) array which is created in an operation system of a network element communication equipment in the embodiment; after the deadlock occurs in the FTP upload task, the socket resources used by the FTP upload task in which the deadlock occurs are placed at the head of the socket resource cycle queue, and after a certain delay, the socket resources, used by the FTP upload task in which the deadlock occurs, are released;

[0027] here, after the initialization is completed, the step that the daemon configured in the network element communication equipment determines whether a deadlock occurs in the FTP upload task through the heartbeat detection mechanism can be implemented in this way:

[0028] step A1: transmitting by the daemon a heartbeat detection message to the FTP upload task periodically, and determining whether a response message (ACK) of the is heartbeat detection message is received within a certain period of time, if the ACK is received, then resetting a heartbeat detection timer, and returning to step A1; otherwise, executing step A2; and

[0029] step A2: adding 1 to a heartbeat failure counter, and determining whether a value of the heartbeat failure counter is greater than a preset value, if the value is greater than the preset value, determining that the deadlock occurs in the FTP upload task; otherwise, executing step A1;

[0030] after determining that the deadlock occurs in the FTP upload task, the daemon can reset the heartbeat failure counter, and step A1 is executed to initiate heartbeat detection in a next cycle;

[0031] the socket resources may comprise a data connection socket, and/or a control connection socket;

[0032] step 102: determining whether the socket resource cycle queue is full, if it is not full, executing step 103; if it is full, executing step 104;

[0033] step 103: putting the socket resources occupied by the current FTP upload task with the deadlock into the socket resource cycle queue, and ending the current heartbeat detection;

[0034] wherein the socket resources occupied by the current FTP upload task with the deadlock are put into the head of the socket resource cycle queue; according to the

FIFO principle, the socket resources in the head of the queue can be released when cycling to the tail of the queue;

[0035] step 104: releasing socket resources which are put into the socket resource cycle queue at the earliest time, and putting the socket resources occupied by the current FTP upload task with the deadlock into the socket resource cycle queue.

[0036] After determining that the deadlock occurs in the FTP upload task, the daemon can reset the heartbeat failure counter, so as to initiate heartbeat detection in a next cycle.

[0037] It should be noted that, a task priority to which the daemon belongs is higher than that of the FTP upload task.

[0038] Correspondingly, the present disclosure further provides an embodiment of a system for FTP deadlock detection and self recovery; as shown in FIG. 2, the system comprises: a daemon 21, an FTP upload task unit 22, a socket resource cycle queue 23 and an FTP server 24, wherein,

[0039] the daemon 21 comprises: a data receiving interface module for collecting data information, a heartbeat detection initiating module with function of initiating heartbeat detection, a module with function of deadlock determination and recovery, a buffer queue managing module with function of managing FTP socket resources, and so on;

[0040] the FTP upload task unit 22 is configured to implement basic operations including establishing a control connection and a data connection with the FTP server and uploading FTP data, and further configured to respond to a heartbeat detection message of the daemon 21 and return a response message, and further to acquire data from the daemon, establish a connection with the FTP and upload the acquired data to the FTP server;

[0041] the socket resource cycle queue 23 is configured to store socket resources used by an FTP upload operation each time a deadlock occurs, such as FTP control connection socket resources and data connection socket resources which are used each time a deadlock occurs; and

[0042] the FTP server 24 is configured to receive data uploaded by the FTP upload task.

[0043] Furthermore, the FTP upload task unit 22 comprises: a heartbeat ACK module, a socket resource reading and writing module with function of reading and storing when socket resources are recovered, and an FTP upload module; wherein the heartbeat ACK module sends direct feedback according to a heartbeat detection message initiated by a heartbeat; however, if an FTP upload operation is being executed at the time, the heartbeat ACK module cannot respond to the heartbeat detection message because it cannot be scheduled; each time initialization is carried out, the socket resource reading and writing module delays releasing according to earliest historical FTP deadlock information which is read by the socket resource cycle buffer queue managing module and socket information related to socket resources; the FTP upload task unit 22 reads data stored by the data receiving message interface, establishes a connection with the FTP server, and at the same time, stores the socket information related to socket resources and writes the socket information into a queue node assigned by the socket resource cycle buffer queue managing module.

[0044] Specifically, the daemon 21 is configured to determine whether a deadlock occurs in the FTP upload task through a heartbeat detection mechanism, if the deadlock occurs, the socket resources used by the FTP upload operation are recorded at the storage location in the assigned socket

resource cycle queue, and an FTP upload task end process is started; the daemon 21 is further configured to determine whether the socket resource cycle queue is full, if it is not full, socket resource information occupied by the current deadlock is put into the socket resource cycle queue, if it is full, the earliest socket resources in the socket resource cycle queue are released, and the socket resource information occupied by the current deadlock is put into the socket resource cycle queue.

[0045] It should be noted that, the daemon 21 is corresponding to a device for FTP deadlock detection and self recovery according to an embodiment of the present disclosure; the device mainly comprises: a heartbeat detection initiating module and a buffer queue managing module, wherein

[0046] the heartbeat detection initiating module is configured to determine whether a deadlock occurs in an FTP upload task through the heartbeat detection mechanism;

[0047] the buffer queue managing module is configured to record socket resources used by an FTP upload operation at the storage location in an assigned socket resource cycle queue, and start an FTP upload task end process; the buffer queue managing module is further configured to determine whether the socket resource cycle queue is full, if it is not full, socket resource information occupied by the current deadlock is put into the socket resource cycle queue, otherwise, the earliest socket resources in the socket resource cycle queue are released, and socket resource information occupied by the current deadlock is put into the socket resource cycle queue.

[0048] In the abovementioned device, the heartbeat detection initiating module comprises: a transmitting module and a determining module, wherein the transmitting module is configured to transmit a heartbeat detection message to an FTP upload task periodically; the determining module is configured to determine whether an ACK of the heartbeat detection message is received within a certain period of time, if the ACK is received, then reset a heartbeat detection timer, and notify the transmitting module to periodically transmit a heartbeat detection message to the FTP upload task, if the ACK is not received, then add 1 to a heartbeat failure counter, and determine whether a value of the heartbeat failure counter is greater than a preset value, if the value is greater than the preset value, then determine the deadlock occurs in the FTP upload task, if the value is not greater than the preset value, the determining module periodically transmits a heartbeat detection message to the FTP upload task.

[0049] In the abovementioned device, the buffer queue managing module can be further configured to notify the determining module to reset the heartbeat failure counter after determining that the deadlock occurs in the FTP upload task.

[0050] Here, it should be explained that, the task priority to which the daemon belongs is higher than that of the FTP upload task.

[0051] The method, device and system for FTP deadlock detection and self recovery which are provided by the present disclosure are described briefly above. The technical solution of the present disclosure is described below with reference to specific embodiments.

Embodiment 1

[0052] As shown in FIG. 3, the main process of implementing FTP deadlock detection and self recovery according to an embodiment of the present disclosure comprises the following steps:

- [0053] step 301: the daemon starts the heartbeat detection timer when initialization is performed;
- [0054] step 302: the daemon transmits a heartbeat detection message to an FTP upload task periodically;
- [0055] step 303: the daemon determines whether an ACK of the last heartbeat detection message is received, if the ACK is received, step 308 is executed, otherwise, step 304 is executed;
- [0056] step 304: the daemon adds 1 to the heartbeat failure counter;
- [0057] step 305: the daemon determines whether an accumulated value of the heartbeat failure counter exceeds a preset value N, if the accumulated value does not exceed N, is then step 301 is executed so as to wait for the next cycle to initiate heartbeat detection, if the accumulated value exceeds N, step 306 is executed;
- [0058] It should be noted that, the heartbeat detection mechanism adopted by the embodiment can monitor a deadlock according to different deadlock determining policies (for example, modifying the value N), instead of taking presence of a semaphore waiting loop as the only basis for determination, thereby not only improving the flexibility of coping with a sudden FTP deadlock on base station equipment, but also decreasing times of resetting a single board in abnormal situation and improving the reliability of the system;
- [0059] step 306: the daemon starts an FTP upload task end process;
- [0060] here, by starting the FTP upload task end process, the daemon completely ends the FTP upload task where the deadlock occurs;
- [0061] step 307: the daemon assigns a new access location in the socket resource cycle queue, initializes the FTP upload task again, resets the heartbeat failure counter, and ends the heartbeat detection initiated in this cycle; and then step 301 is executed, so as to wait for the next cycle to initiate heartbeat detection; and
- [0062] step 308: the daemon resets the heartbeat detection timer, and step 301 is executed, so as to wait for the next cycle to initiate heartbeat detection.

Embodiment 2

- [0063] As shown in FIG. 4, the main process of implementing FTP upload task initialization and data upload according to the present disclosure comprises the following steps:
- [0064] step 401: the daemon triggers an FTP upload task initialization process;
- [0065] wherein the FTP upload task initialization process is triggered by the daemon when initialization or deadlock recovery happens;
- [0066] step 402: the FTP upload task unit determines whether a socket resource cycle queue is full, if it is full, then step 403 is executed; otherwise, step 404 is executed;
- [0067] step 403: resources which are put into the socket resource cycle queue at the earliest time are released, and the socket resources obtained through releasing are used to store socket resource information in this life cycle;
- [0068] step 404: a location of an idle socket resource node is obtained from the socket resource cycle queue to store the socket resource information in this life cycle, then the FTP upload task initialization is completed;

- [0069] step 405: the FTP upload task unit detects whether there is data needed to be uploaded, if there is such data, then step 406 is executed; otherwise, step 408 is executed;
- [0070] step 406: the FTP upload task unit establishes a connection with the FTP server, and records the socket resource information of the current FTP upload operation into the socket resource reading and writing module;
- [0071] wherein the socket resource information of the current FTP upload operation is recorded into the socket resource reading and writing module for temporary storage; when the deadlock occurs in the current FTP upload task, the socket resource information stored in the socket resource reading and writing module is called, and then the socket resource information of the current FTP upload operation is put into the socket resource cycle queue;
- [0072] step 407: data upload is started, after it is completed, the current data FTP upload task is ended, and step 405 is executed; and
- [0073] step 408: the daemon continues waiting and collecting data needed to be uploaded, then step 405 is executed;
- [0074] wherein the data needed to be uploaded comprise various types of information to be reported, such as network element fault data, service data and the like.
- [0075] It should be noted that, the FTP upload task is detected and monitored by a heartbeat detection message sent from the daemon in the whole life cycle all the time.
- [0076] Compared with the prior art, the embodiments of the present disclosure introduce methods such as a method for detecting a deadlock through a heartbeat detection mechanism, a method for releasing socket resources with delay from a socket resource cycle queue when a deadlock occurs, a self recovery method, and the like. Furthermore, the heartbeat detection mechanism can monitor a deadlock according to different deadlock determining policies (for example, modifying a value N), instead of taking presence of a semaphore waiting loop as the only basis for determination, thus not only improving the flexibility of coping with a sudden FTP deadlock of base station equipment, but also decreasing resetting times of a single board in abnormal situation and improving the reliability of the system. In addition, in the embodiments of the present disclosure, control connection socket resources and data connection socket resources are released with delay when a deadlock occurs, and the possibility of presence of a sudden abnormal situation is reduced, thereby ensuring the total amount of occupied socket resources in a base station system are as well as avoiding socket resources in use from overflowing; at the same time, abnormal data receiving and scheduling at a daemon, which is caused by an abnormal FTP upload operation, is prevented, thereby reducing the overstocking risk of all process messages of the task to which the daemon belongs, and playing a role of moderating the high utilization ratio of central processing unit (CPU) of a single board.
- [0077] Obviously, various changes and modifications can be made to the present disclosure by those skilled in the art without departing from the spirit and scope of the present disclosure. Thus, if these modifications and equivalent replacements of the present disclosure are within the scope of the claims of the present disclosure and equal technology thereof, the present disclosure is also intended to include these changes and modifications.

1. A method for File Transfer Protocol (FTP) deadlock detection and self recovery, comprising the following steps:

determining, by a daemon configured in a network element, whether a deadlock occurs in an FTP upload task through a heartbeat detection mechanism, if the deadlock occurs, then recording socket resources used by an FTP upload operation at a storage location in a socket resource cycle queue which is created in the network element, and starting an FTP upload task end process; and

determining whether the socket resource cycle queue is full, if it is not full, then putting socket resource information occupied by the current deadlock into the socket resource cycle queue, if it is full, then releasing earliest socket resources from the socket resource cycle queue, and putting the socket resource information occupied by the current deadlock into the socket resource cycle queue.

2. The method according to claim 1, wherein the step of determining by the daemon whether a deadlock occurs in the FTP upload task through the heartbeat detection mechanism comprises:

A1: transmitting, by the daemon, a heartbeat detection message to the FTP upload task periodically, and determining whether a response message of the heartbeat detection message is received within a certain period of time, if the response message is received, then resetting a heartbeat detection timer, and returning to step A1; otherwise, executing step A2; and

A2: adding 1 to a heartbeat failure counter, and determining whether a value of the heartbeat failure counter is greater than a preset value, if the value is greater than the preset value, then determining that the deadlock occurs in the FTP upload task; otherwise, executing step A1.

3. The method according to claim 2, further comprising: after determining that the deadlock occurs in the FTP upload task, resetting the heartbeat failure counter by the daemon, ending heartbeat detection initiated in this cycle, and returning to step A1 to initiate heartbeat detection in a next cycle.

4. The method according to claim 1, wherein a task priority to which the daemon belongs is higher than that of the FTP upload task.

5. The method according to claim 1, wherein the socket resources comprise a data connection socket, and/or a control connection socket.

6. A device for File Transfer Protocol (FTP) deadlock detection and self recovery, comprising: a heartbeat detection initiating module and a buffer queue managing module, wherein

the heartbeat detection initiating module is configured to determine whether a deadlock occurs in an FTP upload task through a heartbeat detection mechanism; and

the buffer queue managing module is configured to record socket resources used by an FTP upload operation at a storage location in an assigned socket resource cycle queue, and start an FTP upload task end process; and further configured to determine whether the socket resource cycle queue is full, if it is not full, then put socket resource information occupied by the current deadlock into the socket resource cycle queue, if it is full, then release earliest socket resources from the socket resource cycle queue, and is put the socket resource information occupied by the current deadlock into the socket resource cycle queue.

7. The device according to claim 6, wherein the heartbeat detection initiating module comprises: a transmitting module and a determining module, wherein

the transmitting module is configured to transmit a heartbeat detection message to the FTP upload task periodically; and

the determining module is configured to determine whether a response message of the heartbeat detection message is received within a certain period of time, if the response message is received, then reset a heartbeat detection timer, and notify the transmitting module to periodically transmit a heartbeat detection message to the FTP upload task, if the response message is not received, then add 1 to a heartbeat failure counter; and the determining module is further configured to determine whether a value of the heartbeat failure counter is greater than a preset value, if the value is greater than the preset value, then determine that the deadlock occurs in the FTP upload task, otherwise, periodically transmit a heartbeat detection message to the FTP upload task.

8. The device according to claim 7, wherein the buffer queue managing module is further configured to notify the determining module to reset the heartbeat failure counter after determining that the deadlock occurs in the FTP upload task.

9. The device according to claim 6, wherein the socket resources comprise: a data connection socket, and/or a control connection socket;

a task priority to which the daemon belongs is higher than that of the FTP upload task.

10. A system for File Transfer Protocol (FTP) deadlock detection and self recovery, comprising a daemon, an FTP upload task unit, a socket resource cycle queue and an FTP server, wherein

the daemon is configured to determine that a deadlock occurs in an FTP upload task through a heartbeat detection mechanism, and then record socket resources used by an FTP upload operation at a storage location in an assigned socket resource cycle queue, and start an FTP upload task end process; and further configured to determine whether the socket resource cycle queue is full, if it is not full, then put socket resource information occupied by the current deadlock into the socket resource cycle queue, otherwise, release earliest socket resources from the socket resource cycle queue, and put the socket resource information occupied by the current deadlock into the socket resource cycle queue;

the FTP upload task unit is configured to return a heartbeat response in response to the heartbeat detection of the daemon, acquire data from the daemon, establish a connection with the FTP and upload the acquired data to the FTP server;

the socket resource cycle queue is configured to store socket resources used by the FTP upload operation each time a deadlock occurs; and

the FTP server is configured to receive data uploaded by the FTP upload task.

11. The system according to claim 10, wherein the daemon comprises:

a heartbeat detection initiating module configured to determine whether the deadlock occurs in the FTP upload task through the heartbeat detection mechanism; and

a buffer queue managing module configured to record the socket resources used by the FTP upload operation at the storage location in the assigned socket resource cycle queue, and start the FTP upload task end process; and further configured to determine whether the socket resource cycle queue is full, if it is not full, then put the socket resource information occupied by the current

deadlock into the socket resource cycle queue, if it is full, then release the earliest socket resources from the socket resource cycle queue, and put the socket resource information occupied by the current deadlock into the socket resource cycle queue.

* * * * *