TERMINAL DEVICE UPGRADE METHOD, APPARATUS, ELECTRONIC DEVICE, AND STORAGE MEDIUM

CROSS REFERENCE TO RELATED APPLICATIONS

[0001] The present application is a national phase entry of PCT Application No. PCT/CN2024/113798, filed on August 21, 2024, which claims the priority of the Chinese patent application No. 202311572226.4, filed on November 22, 2023, contents of which are incorporated herein by its entireties.

TECHNICAL FIELD

[0002] Embodiments of the present disclosure relate to the technical field of network communication, and more specifically, to a terminal device upgrade method, an apparatus, an electronic device, and a storage medium.

BACKGROUND

[0003] An Over-the-Air (OTA) technology is a technology for remotely managing firmware (also referred to as software) in a terminal device via an air interface of mobile communication. Particularly in the Internet of Things field, since Internet of Things devices may be widely distributed or located at remote regions, by applying an OTA-based software upgrade method, an upgrade cost of IoT terminals may be saved.

[0004] In the related art, by applying the OTA-based software upgrade method, latest version information of target software may be broadcasted to a terminal device, and when the terminal device determines that current version information of the target software is inconsistent with the latest version information, the terminal device may perform software upgrade.

[0005] However, the above software upgrade method may be a forced upgrade method. In a case where the terminal device is processing other service data, competition for resources between software upgrade and service data processing may be caused, or the software upgrade may be performed with priority, such that service functions of the terminal device may not be normally implemented.

SUMMARY

[0006] The present disclosure provides a terminal device upgrade method, an apparatus, an electronic device, and a storage medium, such that software upgrade functions of the terminal device may be implemented in a more rational manner. Technical solution of the present disclosure may be as follows.

[0007] In a first aspect, the present disclosure provides a terminal device upgrade method,

applied to a server and including:

[0008] In response to obtaining a software upgrade request, obtaining a to-be-processed target upgrade task;

[0009] Sending a device status acquisition instruction to a plurality of to-be-upgraded devices associated with the target upgrade task, and receiving device status information returned by each of the plurality of to-be-upgraded devices;

[0010] Determining at least one upgradeable device from the plurality of to-be-upgraded devices based on the device status information returned by each of the plurality of to-be-upgraded devices; and

[0011] Sending a to-be-upgraded software data package to each of the at least one upgradeable device, wherein the upgradeable device is configured to perform software upgrading based on the to-be-upgraded software data package.

[0012] In a second aspect, the present disclosure provides a terminal device upgrade apparatus, which may be a server and may include:

[0013] An obtaining module, configured to, in response to obtaining a software upgrade request, obtain a to-be-processed target upgrade task;

[0014] A transceiver module, configured to: send device status acquisition instruction to a plurality of to-be-upgraded devices associated with the target upgrade task, and receive device status information returned by each of the plurality of to-be-upgraded devices;

[0015] A determination module, configured to determine at least one upgradeable device from the plurality of to-be-upgraded devices based on the device status information returned by each of the plurality of to-be-upgraded devices; and

[0016] A sending module, configured to send a to-be-upgraded software data package to each of the plurality of to-be-upgraded devices, wherein the at least one upgradeable device is configured to perform software upgrading based on the to-be-upgraded software data package.

[0017] In a third aspect, the present disclosure provides an electronic device, including:

[0018] A processor;

[0019] A memory storing a program.

[0020] The program includes instructions. The instructions, when being executed by the processor, are configured to cause the processor to perform the method in the first aspect.

[0021] In a fourth aspect, the present disclosure provides a non-transitory computer-readable storage medium, storing computer instructions. The computer instructions are configured to cause a computer to perform the method in the first aspect.

[0022] According to the terminal device upgrade method, the apparatus, the electronic device,

and the storage medium provided by the embodiments of the present disclosure, in response to obtaining the software upgrade request, the to-be-processed target upgrade task may be obtained. A device status acquisition instruction may be sent to a plurality of to-be-upgraded devices associated with the target upgrade task, and device status information returned by each of the plurality of to-be-upgraded devices may be received. At least one upgradeable device may be determined from the plurality of to-be-upgraded devices based on the device status information returned by each to-be-upgraded device. A to-be-upgraded software data package may be sent to each of the at least one upgradeable device. When a terminal device is performing software upgrade, the device status information of the terminal device may be firstly obtained. When the device status information indicates that the terminal device is upgradeable, the to-be-upgraded software data package may be sent to the terminal device. In this way, forced upgrading of any terminal device in an unsatisfactory condition may be prevented, software upgrade functions of the terminal device may be implemented in a more rational manner.

BRIEF DESCRIPTION OF THE DRAWINGS

[0023] In the following description of exemplary embodiments with reference to the accompanying drawings, more details, features, and advantages of the present disclosure are disclosed.

[0024] FIG. 1 is a schematic view of an exemplary system for performing a terminal device upgrade method according to an embodiment of the present disclosure.

[0025] FIG. 2 is a flow chart of the terminal device upgrade method according to an embodiment of the present disclosure.

[0026] FIG. 3 is a flow chart of an interaction while performing the terminal device upgrade method according to an embodiment of the present disclosure.

[0027] FIG. 4 is a flow chart of another interaction while performing the terminal device upgrade method according to an embodiment of the present disclosure.

[0028] FIG. 5 is a schematic diagram of a terminal device upgrade apparatus according to an embodiment of the present disclosure.

[0029] FIG. 6 is a schematic diagram of a chip according to an embodiment of the present disclosure.

[0030] FIG. 7 is a structural schematic diagram of an electronic device for implementing embodiments of the present disclosure.

DETAILED DESCRIPTIONS

[0031] Embodiments of the present disclosure will be described in more detail below by referring to the accompanying drawings. Although certain embodiments of the present disclosure are shown in the drawings, it shall be understood that the present disclosure may be implemented in various forms and shall not be interpreted as being limited to the embodiments herein. Rather, these embodiments are provided to enable the present disclosure to be understood more thoroughly and completely. It should be understood that the drawings and embodiments of the present disclosure are for illustrative purposes only and are not intended to limit the scope of the present disclosure.

[0032] It should be understood that various steps described in the method of the present disclosure may be performed in different orders and/or in parallel with each other. Furthermore, the method in various embodiments may include additional steps, and/or some shown steps may be omitted. The scope of the present disclosure is not limited thereto.

[0033] The term "include" and variations thereof as used herein are non-exclusive inclusions, i.e., "including but not limited to". The term "based on" is "based at least in part on". The term "an embodiment" means "at least one embodiment"; the term "another embodiment" means "at least one additional embodiment"; the term "some embodiments" means "at least some embodiments". Definitions of other terms will be given in the description below. It should be noted that concepts, such as "first", "second", and so on, mentioned in the present disclosure may only be used to distinguish different devices, modules, or units, and are not intended to limit an order or interdependence of functions performed by the devices, modules, or units.

[0034] It should be noted that limitations based on "a", "a plurality of", and so on mentioned in the present disclosure are illustrative and not restrictive, and any ordinary skilled artisan in the art shall understand that unless the context clearly indicates otherwise, the above limitations shall be interpreted as "one or more".

[0035] Names of messages or information exchanged between a plurality of devices in the embodiments of the present disclosure are for illustrative purposes only and are not intended to limit the scope of the messages or information.

[0036] In the related art, during a process of upgrading software of a terminal device based on the OTA technology, an OTA server may broadcast latest version information of a to-be-upgraded software to a terminal device. After receiving the latest version information of the to-be-upgraded target software, the terminal device, when determining that the latest version information of the target software is inconsistent with current version information of the target software in the terminal device, may directly perform software upgrading. The target software may be any software in the terminal device, such as a device system or a client, which is not limited herein.

[0037] However, the software upgrade method provided in the related art may be a forced upgrade method. When the terminal device is in an idle state, the target software may be reliably upgraded. However, the software upgrade may be forced while the terminal device is processing other service data, such that competition for resources between software upgrade and service data processing may be caused, or the software upgrade may be performed with priority, and service functions of the terminal device may not be normally implemented.

[0038] To solve the above problem, embodiments of the present disclosure may provide a terminal device upgrade method. FIG. 1 shows a schematic diagram of an implementation scenario of a terminal device upgrade system according to an embodiment of the present disclosure. As shown in FIG. 1, the implementation scenario 100 may include a terminal device 101 and a server 102. The terminal device 101 may be a device in a mobile communication network, such as, a terminal device in Internet of Things. The server 102 may be a service device that manages the terminal device. The terminal device may be a computer, a mobile phone, a tablet computer, a wearable device, an IoT terminal, or the like.

[0039] As shown in FIG. 1, when implementing the terminal device upgrade method provided by the embodiments of the present disclosure, the terminal device 101 may establish a wireless or wired link with the server 102 to implement the terminal device upgrade method provided by the embodiments of the present disclosure.

[0040] It should be noted that, in the embodiments of the present disclosure, the server 102 may be a single server or a cluster of a plurality of servers, which is not limited herein. In an example, the server 102 may include a software server and a device management server. The software server may be configured to, in response to obtaining a software upgrade request, obtain a to-be-processed target upgrade task. The device management server may be configured to send a device status acquisition instruction to a plurality of to-be-upgraded devices associated with the target upgrade task; receive device status information returned by each of the plurality of to-be-upgraded devices; and determine at least one upgradeable device from the plurality of to-be-upgraded devices based on the device status information returned by each of the plurality of to-be-upgraded devices. Further, the software server may be configured to send a to-be-upgraded software data package to each upgradeable device. The upgradeable device may be configured to perform software upgrading based on the to-be-upgraded software data package.

[0041] FIG. 2 shows a flow chart of the terminal device upgrade method according to an embodiment of the present disclosure. The method may be applied to a server. As shown in FIG. 2, the method of the embodiment of the present disclosure may include following blocks.

[0042] In a block S201, in response to obtaining the software upgrade request, the to-be-

processed target upgrade task may be obtained.

[0043] In a block S202, the device status acquisition instruction may be sent to the plurality of to-be-upgraded devices associated with the target upgrade task, and the device status information returned by each of the plurality of to-be-upgraded devices may be received.

[0044] In a block S203, at least one upgradeable device may be determined from the plurality of to-be-upgraded devices based on the device status information returned by each of the plurality of to-be-upgraded devices.

[0045] In a block S204, the to-be-upgraded software data package may be sent to each of the at least one upgradeable device.

[0046] The upgradeable device may be configured to perform software upgrading based on the to-be-upgraded software data package.

[0047] In summary, according to the terminal device upgrade method provided by the embodiments of the present disclosure, in response to obtaining the software upgrade request, the to-be-processed target upgrade task may be obtained; the device status acquisition instruction may be sent to the plurality of to-be-upgraded devices associated with the target upgrade task, and the device status information returned by each to-be-upgraded device may be received; at least one upgradeable device may be determined, based on the device status information returned by each to-be-upgraded device, from the plurality of to-be-upgraded devices; the to-be-upgraded software data package may be sent to each upgradeable device. When the terminal device is performing software upgrading, the device status information of the terminal device may be firstly obtained. The to-be-upgraded software data package may be sent to the terminal device only when the device status information indicates that the terminal device is upgradeable. In this way, any terminal device in unsatisfactory conditions may be prevented from performing forced upgrading, software upgrade functions of the terminal device may be performed in a more rational manner.

[0048] A specific implementation of each block in the embodiment shown in FIG. 2 will be described in detail below.

[0049] In a block S201, the server may, in response to obtaining the software upgrade request, obtain the to-be-processed target upgrade task.

[0050] In the present embodiment, the software upgrade request may be sent by a terminal device of a software developer. The software upgrade request may include the to-be-upgraded software data package and a device list of the plurality of to-be-upgraded devices. The to-be-upgraded software data package may be a data package of the to-be-upgraded target software. The to-be-upgraded software data package may include a software package of the target software, as well as a differential package, a patch package, a rollback package, and so on, of the target

software, which is not limited herein. The target software may be any software in the terminal device, such as a device system.

[0051] In an embodiment, a process in which the server obtains the to-be-processed target upgrade task in response to obtaining the software upgrade request, may include the following. The server, in response to obtaining the software upgrade request, may obtain the to-be-upgraded software data package and the device list of the plurality of to-be-upgraded devices; and may create the to-be-processed target upgrade task according to the to-be-upgraded software data package and the device list of the plurality of to-be-upgraded devices.

[0052] In a block S202, the server may send the device status acquisition instruction to the plurality of to-be-upgraded devices associated with the target upgrade task, and receive the device status information returned by each of the plurality of to-be-upgraded devices.

[0053] In the present embodiment, the plurality of to-be-upgraded devices associated with the target upgrade task may include the plurality of to-be-upgraded devices that are to execute the target upgrade task. The device status information may include: whether the device is in an idle state; hardware resource parameters of the to-be-upgraded device, such as a remaining power level and a remaining storage space of the to-be-upgraded device; and an idle state waiting duration of the to-be-upgraded device. The to-be-upgraded device being in the idle state may mean that the to-be-upgraded device is not using any network resource or any hardware resource, such as a state in which the to-be-upgraded device is not processing service data or performing any other software upgrading. The idle waiting duration may be a waiting time length for which the to-be-upgraded device to transits from a non-idle state to the idle state. The idle waiting duration may be estimated by the to-be-upgraded device based on a current service processing speed of the terminal device.

[0054] In an embodiment, a process in which the server sends the device status acquisition instruction to the plurality of to-be-upgraded devices associated with the target upgrade task and receives the device status information returned by each to-be-upgraded device, may include the following. The server may obtain a plurality of device identifiers in the device list carried in the target upgrade task, and send the device status acquisition instruction to the to-be-upgraded device corresponding to each of the plurality of device identifiers. It can be understood that the device status acquisition instruction may be configured to instruct the to-be-upgraded device to return the device status information of the to-be-upgraded device to the server.

[0055] In a block S203, the server may determine at least one upgradeable device from the plurality of to-be-upgraded devices based on the device status information returned by each to-be-upgraded device.

[0056] In the present embodiment, the upgradeable device may be one of the plurality of to-be-upgraded devices that satisfies an upgrade condition. The upgrade condition may include: the to-be-upgraded device being in the idle state. The upgrade condition may further include that a hardware resource of the to-be-upgraded device meets an upgrade requirement. The hardware resource meeting the upgrade requirement may be determined based on practical needs, which is not limited herein. For example, the hardware resource meeting the upgrade requirement may be: a remaining storage space of the to-be-upgraded device being greater than a data size of the to-be-upgraded software data package, and/or the remaining power level of the to-be-upgraded device being greater than or equal to a preset power threshold, and so on. It can be understood that, when the upgrade condition includes the to-be-upgraded device being in the idle state, it may be ensured that the software upgrading may not affect other service processing in the terminal device. When the upgrade condition includes that the hardware resource of the to-be-upgraded device meeting the upgrade requirement, a success rate of the software upgrading may be ensured.

[0057] It should be noted that, in the present embodiment, the server may allow the software developer to upgrade the plurality of to-be-upgraded devices according to different upgrade consistency requirements. The software upgrade request sent by the software developer may further include upgrade consistency information. The upgrade consistency information may be configured to represent a level of upgrade consistency among the plurality of to-be-upgraded devices. When the upgrade consistency information indicates that the plurality of to-be-upgraded devices have a low level of upgrade consistency, the server may indicate that the upgrade task is successfully executed after successfully upgrading the upgradeable device of the plurality of to-be-upgraded devices. When the upgrade consistency information indicates that the plurality of to-be-upgraded devices have a medium level of upgrade consistency, the server may indicate that the upgrade task is successfully executed after successfully upgrading the upgradeable device of the plurality of to-be-upgraded devices and successfully upgrading, with a delay, a non-upgradeable device of the plurality of to-be-upgraded devices. When the upgrade consistency information indicates that the plurality of to-be-upgraded devices have a high level of upgrade consistency, the server may indicate that the upgrade task is successfully executed after successfully upgrading the plurality of to-be-upgraded devices synchronously.

[0058] In an embodiment, a process of determining at least one upgradeable device from the plurality of to-be-upgraded devices based on the device status information returned by each to-be-upgraded device, may include the following. The at least one upgradeable device may be determined from the plurality of to-be-upgraded devices based on the upgrade consistency information in the target upgrade task and the device status information returned by each to-be-

upgraded device. It should be noted that during a process of the server creating the target upgrade task, the server may add the upgrade consistency information obtained from the software upgrade request to the target upgrade task. A device that is capable of performing the software upgrading may be determined from the plurality of to-be-upgraded devices, based on a user requirement for the upgrade consistency of the plurality of to-be-upgraded devices and the device status of each to-be-upgraded device. In this way, other service processing of the to-be-upgraded device may not be affected, diverse requirements for software upgrades may be met, and software upgrading experience for the software developer may be improved.

[0059] In an embodiment, a process of determining the at least one upgradeable device from the plurality of to-be-upgraded devices based on the upgrade consistency information in the target upgrade task and the device status information returned by each to-be-upgraded device, may include the following. When the upgrade consistency information indicates that the plurality of to-be-upgraded devices have the medium level or the low level of upgrade consistency, any of the plurality of to-be-upgraded devices, which may have the device status information indicating that the upgrade condition is satisfied, may be determined as the upgradeable device. When the plurality of to-be-upgraded devices have the medium level and the low level of the upgrade consistency, more than one of the plurality of to-be-upgraded devices that satisfy the upgrade condition may be determined as the upgradeable devices. In this way, software upgrade requirements of the software developer for the medium level or the low level of the upgrade consistency may be met.

[0060] In an embodiment, a process in which the server may determine the at least one upgradeable device from the plurality of to-be-upgraded devices based on the upgrade consistency information in the target upgrade task and the device status information returned by each to-be-upgraded device, may include the following. When the upgrade consistency information indicates that the plurality of to-be-upgraded devices have a high level of upgrade consistency, the server may determine a synchronous upgrading time point for the plurality of to-be-upgraded devices according to the device status information of each to-be-upgraded device, and may determine each to-be-upgraded device as the upgradeable device. When the upgrade consistency of the plurality of to-be-upgraded devices is at the high level, the server may determine the synchronous upgrading time point at which all of the plurality of to-be-upgraded devices are in the idle state based on the device status information of each to-be-upgraded device. In this way, the plurality of to-be-upgraded devices may be upgraded at the synchronous upgrading time point, software upgrade requirements of the software developer for the medium level and the high level of upgrade consistency may be met.

[0061] It should be noted that, in the present embodiment, the plurality of to-be-upgraded devices may all be in the idle state, or all be in the non-idle state; or some of the plurality of to-be-upgraded devices may be in the idle state. A process in which the server may determine the synchronous upgrading time point for the plurality of to-be-upgraded devices according to the device status information of each to-be-upgraded device, may include the following. When the device status information of each to-be-upgraded device indicates that the to-be-upgraded device is in the idle state, a current time point may be determined as the synchronous upgrading time point for the plurality of to-be-upgraded devices. When the device status information of each to-be-upgraded device indicates that the to-be-upgraded device is in the non-idle state, the server may obtain the idle waiting duration from the device status information of each to-be-upgraded device; determine a first longest idle waiting duration of idle waiting durations of all of the plurality of to-be-upgraded devices; and determine a time point obtained by delaying the current time point by the first longest idle waiting duration as the synchronous upgrading time point for the plurality of to-be-upgraded devices. Alternatively, when the device status information of at least one of the plurality of to-be-upgraded devices indicates that the at least one to-be-upgraded device is in the non-idle state, the server may obtain the idle waiting duration from the device status information of the at least one to-be-upgraded device in the non-idle state; determine a second longest idle waiting duration of at least one idle waiting duration of the at least one to-be-upgraded device; and determine a time point obtained by delaying the current time point by the second longest idle waiting duration as the synchronous upgrading time point for the plurality of to-be-upgraded devices.

[0062] In an embodiment, when the software developer has no requirement for the upgrade consistency of the plurality of to-be-upgraded devices, the software upgrade request may not carry the upgrade consistency information. In this case, the process in which the server may determine at least one upgradeable device from the plurality of to-be-upgraded devices based on the device status information returned by each to-be-upgraded device, may include the following. Any of the plurality of to-be-upgraded devices that has the device status information indicating that the upgrade condition is satisfied may be determined as the upgradeable device.

[0063] In a block S204, the server may send the to-be-upgraded software data package to each of the at least one upgradeable device.

[0064] In an embodiment, the upgradeable device may be configured to perform the software upgrading based on the to-be-upgraded software data package. It can be understood that a software upgraded by the upgradeable device may be the target software.

[0065] In an embodiment, a process in which the server sends the to-be-upgraded software data

package to each upgradeable device, may include the following. The server may send the to-be-upgraded software data package to each upgradeable device based on a Message Queuing Telemetry Transport (MQTT) protocol. It can be understood that, in the present embodiment, during a deployment phase of deploying terminal devices, the server may configure the terminal devices to subscribe to a target topic. In this way, during a software upgrade phase, the server may publish the to-be-upgraded software data package to the target topic and synchronize the to-be-upgraded software data package to the upgradeable device. The target topic may be an identifier of the target software. Since the terminal device and the server may be in a long-live connection state during transmitting the software data package based on the MQTT protocol, a resource overhead during synchronizing the software data package from the server to the terminal device may be reduced, and a data package transmission efficiency may be improved.

[0066] It should be noted that, in the present embodiment, when the upgrade consistency information indicates that the plurality of to-be-upgraded devices have the high level of upgrade consistency, the process in which the server sends the to-be-upgraded software data package to each upgradeable device, may include the following. The server may send the to-be-upgraded software data package to each upgradeable device at the synchronous upgrading time point. In this way, in scenarios requiring the high level of upgrade consistency, the plurality of to-be-upgraded devices may be synchronously upgraded.

[0067] It should also be noted that, in the embodiments of the present disclosure, when the plurality of to-be-upgraded devices have the medium level of upgrade consistency, the non-upgradeable device of the plurality of to-be-upgraded devices may need to be upgraded after a delay. In this case, in an embodiment, when the upgrade consistency information indicates that the plurality of to-be-upgraded devices have the medium level of upgrade consistency, the server may determine an upgrade delay duration for a delayed upgrade device according to the device status information of the delayed upgrade device of the plurality of to-be-upgraded devices. Further, the server may send a delayed upgrade instruction generated based on the upgrade delay duration to the delayed upgrade device. The delayed upgrade device may be any non-upgradeable device of the plurality of to-be-upgraded devices. The delayed upgrade instruction may be configured to instruct the delayed upgrade device to perform, after delaying for the upgrade delay duration, the software upgrading according to the to-be-upgraded software data package sent by the server. It can be understood that, after the upgrade delay duration, the server may send the to-be-upgraded software data package to each delayed upgrade device. In this way, when the upgrade consistency of the plurality of to-be-upgraded devices is at the medium level, any of the plurality of to-be-upgraded devices that cannot be upgraded at the current time point may be determined

as the delayed upgrade device, and the delayed upgrade device may perform upgrading after a delay, such that software upgrade requirements of the software developer under the medium level of upgrade consistency may be met.

[0068] A process in which the server determines the upgrade delay duration for the delayed upgrade device according to the device status information of the delayed upgrade device of the plurality of to-be-upgraded devices, may include the following. For each delayed upgrade device, when the device status information indicates that the delayed upgrade device is in the non-idle state, the server may determine the idle state waiting duration of the delayed upgrade device as the upgrade delay duration. Alternatively, when the device status information indicates that the remaining power level of the delayed upgrade device is less than the preset power threshold, the server may determine a time difference between the current time point and a time point at which first feedback information is obtained as the upgrade delay duration. The first feedback information may indicate that the remaining power level of the delayed upgrade device is greater than or equal to the preset power threshold. Alternatively, when the device status information indicates that the remaining storage space of the delayed upgrade device is less than or equal to the data size of the to-be-upgraded software data package, the server may determine a time difference between the current time point and a time point at which second feedback information is obtained as the upgrade delay duration. The second feedback information may indicate that the remaining storage space of the delayed upgrade device is greater than the data size of the to-be-upgraded software data package.

[0069] It should be noted that, in the embodiments of the present disclosure, when a plurality of upgradeable terminal devices are determined, scenarios where the plurality of upgradeable devices need to be upgraded sequentially may occur. For example, the terminal devices may access the network via a base station to establish links with the server, and the number of devices that each base station can serve simultaneously may be limited. Therefore, when the plurality of upgradeable devices are present, the plurality of upgradeable devices may be upgraded in various batches. During upgrading one batch of terminal devices, a next batch of to-be-upgraded terminal devices may be notified to wait, such that the plurality of terminal devices may perform upgrading more efficiently and orderly. Alternatively, dependencies among the plurality of upgradeable devices may be present. During upgrading the plurality of upgradeable devices, remaining upgradeable devices that depend on the upgrading upgradeable device may be notified to wait, such that the plurality of terminal devices having the dependencies may perform upgrading more efficiently and orderly.

[0070] In an embodiment, when a plurality of upgradeable devices are determined, before

sending the to-be-upgraded software data package to each upgradeable device, the server may determine an upgrade duration for each upgradeable device based on historical upgrade information. The process in which the server sends the to-be-upgraded software data package to each upgradeable device may include the following. The server may send the to-be-upgraded software data package to a first upgradeable device of the plurality of upgradeable devices and may send a first delayed upgrade instruction to a second upgradeable device of the plurality of upgradeable devices. The first delayed upgrade instruction may be generated based on a first upgrade duration of the first upgradeable device, and the first delayed upgrade instruction may be configured to instruct the second upgradeable device to perform, after delaying for the first upgrade duration, software upgrading according to the to-be-upgraded software data package sent by the server. In an order the plurality of upgradeable devices performing upgrading, the second upgradeable device may perform upgrading later than the first upgradeable device. For the plurality of upgradeable devices that may perform upgrading following an order, after sending the to-be-upgraded software data package to the first upgradeable device of the plurality of upgradeable devices, the delayed upgrade instruction may be sent to the second upgradeable device which is to perform upgrading after the first upgradeable device according to the order. In this way, the second upgradeable device may be prepared for performing upgrading, and the plurality of upgradeable devices may perform upgrading in a more orderly manner.

[0071] The historical upgrade information may be an upgrade information parameter during at least one historical upgrade process within a past period. The upgrade information parameter may be data of a software data package size, data of a network condition, and the upgrade duration, and so on. The process in which the server determines the upgrade duration for each upgradeable device based on the historical upgrade information may include the following. For each upgradeable device, the server may determine a first ratio of a historical software data package size to the size of the to-be-upgraded software data package; determine a second ratio of the data of the historical network condition to data of a current network condition; determine a product of the first ratio and the second ratio to obtain a time ratio parameter; and determine a product of the time ratio parameter and the historical upgrade duration to obtain the upgrade duration for each upgradeable device.

[0072] It should be noted that, in the embodiments of the present disclosure, the historical upgrade information may be information from the historical upgrade process of the terminal device located in a region where the upgradeable device is located, or may be information from the historical upgrade process of the terminal device located in a region where the upgradeable device is located. In a case where the historical information includes information of a plurality of

historical upgrade processes, the upgrade duration determined for the upgradeable device based on each historical upgrade process may be determined, and an average of upgrade durations determined based on the plurality of historical upgrade processes may be determined as the upgrade duration for the upgradeable device.

[0073] It can be understood that in a case where the plurality of upgradeable devices are upgraded in various batches, the first upgradeable device is an upgradeable device in a first set of upgradeable devices of a plurality of sets of upgradeable devices determined based on the plurality of upgradeable devices. The second upgradeable device is an upgradeable device in a second set of upgradeable devices of the plurality of sets of upgradeable devices. The upgradeable device in the second set of upgradeable devices may perform upgrading later than the upgradeable device in the first set of upgradeable devices. It should be noted that, in the embodiments of the present disclosure, a rule for grouping the plurality of upgradeable devices into the various batches and an order in which the plurality of sets of upgradeable devices performing upgrading may be determined based on practical needs, which is not limited herein. In an example, the plurality of upgradeable devices may be randomly grouped into the plurality of sets of upgradeable devices, and each set of upgradeable devices may have a same number of upgradeable devices. The first set of upgradeable devices may be any one set of the plurality of sets of upgradeable devices, and the second set of upgradeable devices may be any one set of the plurality of sets of upgradeable devices. The first set of upgradeable devices and the second set of upgradeable devices may be different sets.

[0074] In an embodiment, as shown in FIG. 3, FIG. 3 shows an interaction diagram of a terminal device upgrade method. After the server sends the to-be-upgraded software data package to the first upgradeable device of the plurality of upgradeable devices and sends the first delayed upgrade instruction to the second upgradeable device, and when the server receives an upgrade pause request sent by the first upgradeable device during sending the to-be-upgraded software data package to the first upgradeable device, the server may pause sending a fragment of the to-be-upgraded software data package to the first upgradeable device; and send a fragment of the to-be-upgraded software data package to the second upgradeable device. After receiving a continue sending instruction sent by the first upgradeable device, the server may pause sending the fragment of the to-be-upgraded software data package to the second upgradeable device, and may send a remaining fragment of the to-be-upgraded software data package to the first upgradeable device. Further, after determining that the first upgradeable device completes performing the upgrading, the server may send a remaining fragment of the to-be-upgraded software data package to the second upgradeable device. A terminal device that is performing upgrading may be allowed

to pause performing the software upgrading, such that the terminal device that is performing the upgrading can process more urgent service data than the software upgrading, ensuring that service processing can be performed normally. Furthermore, after the terminal device that is performing upgrading pauses performing upgrading, the to-be-upgraded software data package may be sent to the terminal device, which is waiting for performing upgrading and is to perform upgrading later than the paused terminal device according to the order, and the to-be-upgraded software data package may be sent to the waiting terminal device during a pause period. In this way, an upgrading efficiency may be improved, while the order of the terminal devices performing upgrading is ensured.

[0075] In an embodiment, as shown in FIG. 4, FIG. 4 shows an interaction flow chart of the terminal device upgrade method. Dependencies are present among the plurality of upgradeable devices. The first upgradeable device may be a source-end upgradeable device of the plurality of upgradeable devices with the dependencies. The source-end upgradeable device is one of the plurality of upgradeable devices with the dependencies that is not dependent upon any other device. After the server sends the to-be-upgraded software data package to the first upgradeable device of the plurality of upgradeable devices and sends the first delayed upgrade instruction to the second upgradeable device, the server may further perform the following. When the server determines that an actual upgrade duration of the first upgradeable device is less than a first upgrade duration and the second upgradeable device is in a sleep state, the server may send, after the source-end upgradeable device completes performing upgrading, a fragment of the to-be-upgraded software data package to a third upgradeable device of the plurality of upgradeable devices that depends on the second upgradeable device. After a target duration, the server may send the to-be-upgraded software data package to the second upgradeable device; and send a second delayed upgrade instruction, which is generated based on a second upgrade duration of the second upgradeable device, to the third upgradeable device. The second delayed upgrade instruction may be configured to instruct the third upgradeable device to perform, after delaying for the second upgrade duration, software upgrading according to the to-be-upgraded software data package sent by the server. The target duration is a difference between the first upgrade duration and the actual upgrade duration of the first upgradeable device. For the plurality of upgradeable devices having the dependencies, in a case where the terminal device that is performing upgrading finishes the upgrading early and a device depending on the early-finished terminal device is in the sleep state, the server may send the to-be-upgraded software data package to a next dependent device. In this way, the to-be-upgraded software data package may be sent to the waiting terminal device during a time period when the device is in the sleep state. The upgrade

efficiency of the plurality of upgradeable devices with the dependencies may be improved, and the order in which the plurality of upgradeable devices performing upgrading may be ensured.

[0076] After receiving the first delayed upgrade instruction, the second upgradeable device may start waiting to receive the to-be-upgraded software data package sent by the server, so as to perform the software upgrading. However, when a waiting duration exceeds a preset duration set by the user, the second upgradeable device may enter the sleep state to reduce energy consumption of the second upgradeable device. It can be understood that, after the second upgradeable device receives the first delayed upgrade instruction, and when the second upgradeable device enters the sleep state, the second upgradeable device may automatically wake up to perform upgrading after delaying for the first upgrade duration from a time point of receiving the first delayed upgrade instruction. After the second upgradeable device completes performing the upgrading, the server may continue to send the remaining fragment of the to-be-upgraded software data package to the third upgradeable device, such that the third upgradeable device may perform upgrading. It can be understood that, for the plurality of upgradeable devices with the dependencies, only after one upgradeable device performs upgrading successfully, a next upgradeable device that depends on the one upgradeable device may perform upgrading, however, the server sending the to-be-upgraded software data package to each upgradeable device may not be affected.

[0077] It should be noted that, in the embodiments of the present disclosure, for each upgradeable device, after obtaining the to-be-upgraded software data package, the to-be-upgraded software data package may be verified based on verification information sent by the server. When verification is successful, the upgradeable device may complete the upgrading, may restart, and may update a software version number. Furthermore, the upgradeable device may send upgrade an upgrade success information to the server, such that the server may determine that an upgrade status of the upgradeable device is that upgrading is performed successfully.

[0078] The above description mainly introduces technical solutions provided by the embodiments of the present disclosure from the perspective of the server. It can be understood that, in order to implement the above functions, the server may include corresponding hardware structures and/or software modules for performing each function. Any ordinary skilled artisan in the art shall understand that, in combination with units and algorithm blocks of the examples described in the embodiments disclosed herein, the present disclosure may be implemented in hardware or in a combination of hardware and computer software. Whether a function is performed by hardware or by computer software driving hardware, may depend on a specific application and design constraints of the technical solution. Any ordinary skilled artisan in the art may perform different methods for various applications to implement the described functions, and

such implementation shall not be considered beyond the scope of the present disclosure.

[0079] The embodiments of the present disclosure may perform functional unit division for the server according to the above method examples. For example, each functional module may be determined corresponding to each function, or two or more functions may be integrated into one processing module. The above integrated module may be implemented in a form of hardware or in a form of software functional modules. It should be noted that division of modules in the embodiments of the present disclosure may be illustrative and may be determined based on merely a logical function, and there may be other division manners in practical implementation.

[0080] In a case where each functional module is determined corresponding to each function, embodiments of the present disclosure may provide a terminal device upgrade apparatus. The terminal device upgrade apparatus may be a server or a chip applied to the server. FIG. 5 shows a schematic block diagram of functional modules of the terminal device upgrade apparatus according to an embodiment of the present disclosure. As shown in FIG. 5, the terminal device upgrade apparatus 500 may include the following.

[0081] An obtaining module 501 may be configured to, in response to obtaining the software upgrade request, obtain the to-be-processed target upgrade task.

[0082] A transceiver module 502 may be configured to send the device status acquisition instruction to the plurality of to-be-upgraded devices associated with the target upgrade task, and to receive the device status information returned by each to-be-upgraded device.

[0083] A determination module 503 may be configured to determine at least one upgradeable device from the plurality of to-be-upgraded devices based on the device status information returned by each to-be-upgraded device.

[0084] A sending module 504 may be configured to send the to-be-upgraded software data package to each of the at least one upgradeable device, and the upgradeable device may be configured to perform software upgrade based on the to-be-upgraded software data package.

[0085] In some embodiments, the determination module 503 may be configured to:

[0086] Determine the at least one upgradeable device from the plurality of to-be-upgraded devices based on the upgrade consistency information in the target upgrade task and the device status information returned by each to-be-upgraded device. The upgrade consistency information may be configured to represent a level of upgrade consistency of the plurality of to-be-upgraded devices.

[0087] In some embodiments, the determination module 503 may be configured to perform the following.

[0088] When the upgrade consistency information indicates that the plurality of to-be-upgraded

devices have a low level or a medium level of upgrade consistency, the determination module 503 may be configured to determine the to-be-upgraded device, which has the device status information indicating that the upgrade condition is satisfied, as the upgradeable device.

[0089] In some embodiments, the apparatus may further include a first delay processing module 505, configured to perform the following.

[0090] The first delay processing module 505 may be configured to determine the upgrade delay duration for the delayed upgrade device according to device status information of the delayed upgrade device of the plurality of to-be-upgraded devices. The delayed upgrade device may be the non-upgradeable device of the plurality of to-be-upgraded devices.

[0091] The first delay processing module 505 may be configured to send the delayed upgrade instruction generated based on the upgrade delay duration to the delayed upgrade device. The delayed upgrade instruction may be configured to instruct the delayed upgrade device to perform, after delaying for the upgrade delay duration, software upgrading according to the to-be-upgraded software data package sent by the server.

[0092] In some embodiments, the determination module 503 may be configured to perform the following.

[0093] When the upgrade consistency information indicates that the plurality of to-be-upgraded devices have the high level of upgrade consistency, the determination module 503 may be configured to determine the synchronous upgrading time point for the plurality of to-be-upgraded devices according to the device status information of each to-be-upgraded device; and determine each to-be-upgraded device as the upgradeable device.

[0094] Sending the to-be-upgraded software data package to each upgradeable device may include the following.

[0095] The to-be-upgraded software data package may be sent to each upgradeable device at the synchronous upgrading time point.

[0096] In some embodiments the apparatus may further include a duration determination module 506 configured to perform the following.

[0097] The duration determination module 506 may be configured to determine the upgrade duration for each upgradeable device based on the historical upgrade information.

[0098] Sending the to-be-upgraded software data package to each upgradeable device may include the following.

[0099] The to-be-upgraded software data package may be sent to the first upgradeable device of the plurality of upgradeable devices, and the first delayed upgrade instruction may be sent to the second upgradeable device. The first delayed upgrade instruction may be generated based on

the first upgrade duration of the first upgradeable device. The first delayed upgrade instruction may be configured to instruct the second upgradeable device to perform, after delaying for the first upgrade duration, software upgrading according to the to-be-upgraded software data package sent by the server. In the order of the plurality of upgradeable devices performing upgrading, the second upgradeable device of the plurality of upgradeable devices performs upgrading later than the first upgradeable device.

[00100] In some embodiments, the apparatus may further include a second delay processing module 507 configured to perform the following.

[00101] When receiving the upgrade pause request sent by the first upgradeable device during sending the to-be-upgraded software data package to the first upgradeable device, the second delay processing module 507 may be configured to pause sending the fragment of the to-be-upgraded software data package to the first upgradeable device, and send the fragment of the to-be-upgraded software data package to the second upgradeable device.

[00102] After receiving the continue sending instruction sent by the first upgradeable device, the second delay processing module 507 may be configured to pause sending the fragment of the to-be-upgraded software data package to the second upgradeable device, and send the remaining fragment of the to-be-upgraded software data package to the first upgradeable device.

[00103] After determining that the first upgradeable device completes performing upgrading, the second delay processing module 507 may be configured to send the remaining fragment of the to-be-upgraded software data package to the second upgradeable device.

[00104] In some embodiments, dependencies are present among the plurality of upgradeable devices, and the first upgradeable device may be the source-end upgradeable device among the plurality of upgradeable devices with the dependencies. The apparatus may further include a third delay processing module 508 configured to perform the following.

[00105] When determining that the actual upgrade duration of the first upgradeable device is less than the first upgrade duration and the second upgradeable device is in the sleep state, the third delay processing module 508 configured to send, after the source-end upgradeable device completes performing upgrading, the fragment of the to-be-upgraded software data package to the third upgradeable device of the plurality of upgradeable devices that depends on the second upgradeable device.

[00106] After the target duration, the third delay processing module 508 configured to send the to-be-upgraded software data package to the second upgradeable device, and send a second delayed upgrade instruction generated based on the second upgrade duration of the second upgradeable device to the third upgradeable device. The second delayed upgrade instruction may

be configured to instruct the third upgradeable device to perform, after delaying for the second upgrade duration, software upgrading according to the to-be-upgraded software data package sent by the server. The target duration may be the difference between the first upgrade duration and the actual upgrade duration of the first upgradeable device.

[00107] FIG. 6 shows a schematic block diagram of a chip according to an embodiment of the present disclosure. As shown in FIG. 6, the chip 600 may include one or more (including two) processors 601 and a communication interface 602. The communication interface 602 may support the server in performing data transceiving blocks in the above terminal device upgrade method. The processor 601 may support the server in performing data processing blocks in the above terminal device upgrade method.

[00108] In some embodiments, as shown in FIG. 6, the chip 600 may further include a memory 603. The memory 603 may include read-only memory and a random access memory, and may provide operation instructions and data to the processor. A portion of the memory may further include a non-volatile random access memory (NVRAM).

[00109] In some embodiments, as shown in FIG. 6, the processor 601 may execute corresponding operations by invoking operation instructions stored in the memory (the operation instructions may be stored in an operating system). The processor 601 may control the processing operations of any one of the terminal devices, and the processor may be referred to as a central processing unit (CPU). The memory 603 may include a read-only memory and a random access memory, and may provide instructions and data to the processor 601. A portion of the memory 603 may further include the NVRAM. For example, in an application, the memory, the communication interface, and the memory may be coupled to each other via a bus system. The bus system may include, in addition to a data bus, a power bus, a control bus, and a status signal bus. However, for clarity of illustration, various buses may be labeled as a bus system 604 in FIG. 6.

[00110] The methods disclosed in the above embodiments of the present disclosure may be applied to the processor or implemented by the processor. The processor may be an integrated circuit chip having signal processing capabilities. In an implementation process, the blocks of the above methods may be completed by hardware integrated logic circuits or software instructions in the processor. The above processor may be a general-purpose processor, a digital signal processor (DSP), an ASIC, a field-programmable gate array (FPGA) or any other programmable logic device, a discrete gate or a transistor logic device, or a discrete hardware component. The processor may implement or execute the methods, steps, and logic block diagrams disclosed in the embodiments of the present disclosure. The general-purpose processor may be a

microprocessor, or the processor may be any conventional processor or the like. The blocks of the methods disclosed in the embodiments of the present disclosure may be directly embodied as being executed and completed by a hardware decoding processor, or executed and completed by a combination of hardware and software modules in a decoding processor. The software module may be located in a mature storage medium in the art, such as a random access memory, a flash memory, a read-only memory, a programmable read-only memory, an electrically erasable programmable memory, or a register. The storage medium may be located in the memory, and the processor may read information from the memory and complete the blocks of the above methods in combination with hardware thereof.

[00111] Embodiments of the present disclosure may further provide an electronic device, including: at least one processor; and a memory communicatively connected to the at least one processor. The memory may store a computer program executable by the at least one processor. The computer program, when executed by the at least one processor, may cause the electronic device to perform the method according to the embodiments of the present disclosure.

[00112] Embodiments of the present disclosure may further provide a non-transitory computer-readable storage medium storing a computer program. The computer program, when executed by a processor of a computer, may cause the computer to perform the method according to the embodiments of the present disclosure.

[00113] Embodiments of the present disclosure may further provide a computer program product including a computer program. The computer program, when executed by a processor of a computer, may cause the computer to perform the method according to the embodiments of the present disclosure.

[00114] As shown in FIG. 7, a structural block diagram of an electronic device that can be used as a server of the present disclosure will be described. The electronic device may be an example of a hardware device applicable to various aspects of the present disclosure. The electronic device may be intended to represent various forms of digital electronic computer devices, such as a laptop computer, a desktop computer, a workstation, a personal digital assistant, a server, a blade server, a mainframe computer, and any other suitable computer. The electronic device may further represent various forms of mobile devices, such as a personal digital assistant, a cellular phone, a smart phone, a wearable device, and any other similar computing device. Components shown herein, their connections and relationships, and their functions, are merely examples and are not intended to limit implementation of the present disclosure described and/or claimed herein.

[00115] As shown in FIG. 7, an electronic device 700 may include a computing unit 701 that can perform various appropriate actions and processes according to a computer program stored in

a read-only memory (ROM) 702 or a computer program loaded from a storage unit 708 into a random access memory (RAM) 703. In the RAM 703, various programs and data required for the operation of the electronic device 700 may be stored. The computing unit 701, the ROM 702, and the RAM 703 may be connected to each other via a bus 704. An input/output (I/O) interface 705 may be connected to the bus 704.

[00116] A plurality of components in the electronic device 700 may be connected to the I/O interface 705, and the plurality of components may include: an input unit 706, an output unit 707, a storage unit 708, and a communication unit 709. The input unit 706 may be any type of device capable of inputting information to the electronic device 700. The input unit 706 may receive input numeric or character information and generate key signal inputs related to user settings and/or function control of the electronic device. The output unit 707 may be any type of device capable of presenting information, and may include, but not limited to: a display, a speaker, a video/audio output terminal, a vibrator, and/or a printer. The storage unit 708 may include, but not limited to: a magnetic disk and an optical disk. The communication unit 709 may allow the electronic device 700 to exchange information/data with other devices through a computer network such as an Internet and/or various telecommunication network, and may include, but not limited to: a modem, a network card, an infrared communication device, a wireless communication transceiver and/or a chipset, such as a Bluetooth™ device, a WiFi device, a WiMax device, a cellular communication device, and/or the like.

[00117] The computing unit 701 may be various general-purpose and/or special-purpose processing components having processing and computing capabilities. Some examples of the computing unit 701 may include, but not limited to: a central processing unit (CPU), a graphics processing unit (GPU), various dedicated artificial intelligence (AI) computing chips, various computing units running machine learning model algorithms, a digital signal processor (DSP), and any appropriate processor, controller, microcontroller, and so on. The computing unit 701 may perform the various methods and processes described above. For example, in some embodiments, the method according to the embodiments of the present disclosure may be implemented as a computer software program tangibly embodied in a machine-readable medium, such as the storage unit 708. In some embodiments, part or all of the computer program may be loaded and/or installed onto the electronic device 700 via the ROM 702 and/or the communication unit 709. In some embodiments, the computing unit 701 may be configured in any other suitable manner (such as by means of firmware) to perform the method according to the embodiments of the present disclosure.

[00118] Program codes for implementing the methods of the present disclosure may be written

in any combination of one or more programming languages. The program codes may be provided to a processor or a controller of a general-purpose computer, a special-purpose computer, or any other programmable data processing apparatus, such that the program codes, when executed by the processor or controller, may cause the functions/operations specified in the flow chart and/or the block diagram to be implemented. The program codes may execute entirely on the machine, partially on the machine, as a stand-alone software package, partially on the machine and partially on a remote machine, or entirely on the remote machine or server.

[00119] In the context of the present disclosure, a machine-readable medium may be a tangible medium that may contain or store a program to be used by or in connection with an instruction execution system, apparatus, or device. The machine-readable medium may be a machine-readable signal medium or a machine-readable storage medium. The machine-readable medium may include, but not limited to: an electronic, magnetic, optical, electromagnetic, infrared, or semiconductor system, apparatus, or device, or any suitable combination of the foregoing. More specific examples of the machine-readable storage medium may include electrical connection based on one or more wires, a portable computer disk, a hard disk, a random access memory (RAM), a read-only memory (ROM), an erasable programmable read-only memory (EPROM or flash memory), an optical fiber, a portable compact disc read-only memory (CD-ROM), an optical storage device, a magnetic storage device, or any suitable combination of the foregoing.

[00120] As used herein, the terms “machine-readable medium” and “computer-readable medium” may refer to any computer program product, device, and/or apparatus (such as a magnetic disk, an optical disk, a memory, a programmable logic device (PLD)) for providing machine instructions and/or data to a programmable processor, including a machine-readable medium that receives machine instructions as a machine-readable signal. The term “machine-readable signal” may refer to any signal used to provide machine instructions and/or data to a programmable processor.

[00121] In order to provide interaction with a user, the systems and techniques described here may be implemented on a computer having: a display device (such as a cathode ray tube (CRT) monitor or a liquid crystal display (LCD) monitor) for displaying information to the user; and a keyboard and a pointing device (such as a mouse or a trackball) through which the user may provide input to the computer. Other kinds of devices may be used to provide interaction with the user. For example, feedback provided to the user may be any form of sensory feedback (such as visual feedback, auditory feedback, or tactile feedback); and input from the user may be received in any form (including acoustic input, speech input, or tactile input).

[00122] The systems and techniques described here can be implemented in a computing system

that includes a backend component (such as a data server), or a computing system that includes a middleware component (such as an application server), or a computing system that includes a frontend component (such as a user computer having a graphical user interface or a web browser through which a user can interact with an implementation of the systems and techniques described here), or any combination of the backend component, the middleware component, or the frontend component. The components of the system may be interconnected by any form or medium of digital data communication (such as a communication network). Examples of communication networks may include: a local area network (LAN), a wide area network (WAN), and the Internet.

[00123] The computer system may include clients and servers. Clients and servers are generally remote from each other and typically interact with each other through a communication network. The relationship of a client and a server arises by virtue of computer programs running on the respective computers and having a client-server relationship to each other.

[00124] In the above embodiments, all or part of the implementation may be implemented by software, hardware, firmware, or any combination thereof. When implemented by software, all or part may be implemented in the form of a computer program product. The computer program product may include one or more computer programs or instructions. When the computer programs or instructions are loaded and executed on a computer, processes or functions described in the embodiments of the present disclosure may be all or partially executed. The computer may be a general-purpose computer, a special-purpose computer, a computer network, a terminal, a user device, or any other programmable apparatus. The computer programs or instructions may be stored in a computer-readable storage medium or transmitted from one computer-readable storage medium to another computer-readable storage medium. For example, the computer programs or instructions may be transmitted from one website site, one computer, one server, or one data center to another website site, another computer, another server, or another data center by in a wired or wireless manner. The computer-readable storage medium may be any available medium that can be accessed by a computer or a data storage device such as a server or data center that integrates one or more available media. The available medium may be a magnetic medium, such as a floppy disk, a hard disk, a magnetic tape; an optical medium, such as a digital video disc (DVD); or a semiconductor medium, such as a solid-state drive (SSD).

[00125] Although the present disclosure is described by referring to specific features and embodiments thereof, obviously, various modifications and combinations may be made without departing from the spirit and scope of the present disclosure. Accordingly, the specification and drawings are merely exemplary illustrations of the present disclosure as defined by the appended claims, and are considered to cover any and all modifications, variations, combinations, or

equivalents within the scope of the present disclosure. Obviously, various modifications and variations made to the present disclosure by any ordinary skilled artisan in the art are not departing from the spirit and scope of the present disclosure. Therefore, when the modifications and variations of the present disclosure fall within the scope of the claims of the present disclosure and equivalents thereof, the present disclosure is also intended to include the modifications and variations.

What is claimed is:

1. A terminal device upgrade method, applied to a server, the method comprising:

in response to obtaining a software upgrade request, obtaining a to-be-processed target upgrade task;

sending a device status acquisition instruction to a plurality of to-be-upgraded devices associated with the target upgrade task, and receiving device status information returned by each of the plurality of to-be-upgraded devices;

determining at least one upgradeable device from the plurality of to-be-upgraded devices based on the device status information returned by each of the plurality of to-be-upgraded devices;

sending a to-be-upgraded software data package to each of the at least one upgradeable device, wherein the upgradeable device is configured to perform software upgrading based on the to-be-upgraded software data package.

2. The terminal device upgrade method of claim 1, wherein, the determining at least one upgradeable device from the plurality of to-be-upgraded devices based on the device status information returned by each of the plurality of to-be-upgraded devices, comprises:

determining the at least one upgradeable device from the plurality of to-be-upgraded devices based on upgrade consistency information in the target upgrade task and the device status information returned by each of the to-be-upgraded devices, wherein the upgrade consistency information is configured to represent a level of upgrade consistency among the plurality of to-be-upgraded devices.

3. The terminal device upgrade method of claim 2, wherein, the determining the at least one upgradeable device from the plurality of to-be-upgraded devices based on upgrade consistency information in the target upgrade task and the device status information returned by each of the to-be-upgraded devices, comprises:

in response to the upgrade consistency information indicating that the plurality of to-be-upgraded devices have a low level or a medium level of upgrade consistency, determining at least one to-be-upgraded device, which has the device status information indicating that an upgrade condition is satisfied, from the plurality of to-be-upgraded devices as the at least one upgradeable device.

4. The terminal device upgrade method of claim 3, wherein, in response to the upgrade consistency information indicating that the plurality of to-be-upgraded devices have the medium level of upgrade consistency, the terminal device upgrade method further comprises:

determining an upgrade delay duration for a delayed upgrade device according to the device

status information of the delayed upgrade device of the plurality of to-be-upgraded devices, wherein the delayed upgrade device is a non-upgradeable device of the plurality of to-be-upgraded devices;

sending a delayed upgrade instruction generated based on the upgrade delay duration to the delayed upgrade device, wherein the delayed upgrade instruction is configured to instruct the delayed upgrade device to perform, after delaying for the upgrade delay duration, software upgrading according to the to-be-upgraded software data package sent by the server.

5. The terminal device upgrade method of claim 2, wherein,

the determining the at least one upgradeable device from the plurality of to-be-upgraded devices based on upgrade consistency information in the target upgrade task and the device status information returned by each of the to-be-upgraded devices, comprises:

in response to the upgrade consistency information indicating that the plurality of to-be-upgraded devices have a high level of upgrade consistency, determining a synchronous upgrading time point for the plurality of to-be-upgraded devices according to the device status information of each of the plurality of to-be-upgraded devices, and determining each of the plurality of to-be-upgraded devices as the upgradeable device;

the sending a to-be-upgraded software data package to each of the at least one upgradeable device, comprises:

sending the to-be-upgraded software data package to each upgradeable device at the synchronous upgrading time point.

6. The terminal device upgrade method of claim 5, wherein, the terminal device upgrade method further comprises:

in a case where the device status information of each of the plurality of to-be-upgraded devices indicates that the to-be-upgraded device is in an idle state, determining a current time point as the synchronous upgrading time point for the plurality of to-be-upgraded devices.

7. The terminal device upgrade method of claim 5, wherein, the terminal device upgrade method further comprises:

in a case where the device status information of each of the plurality of to-be-upgraded devices indicates that the to-be-upgraded device is in a non-idle state, obtaining an idle waiting duration from the device status information of each of the plurality of to-be-upgraded devices; determining a longest idle waiting duration of all of the plurality of to-be-upgraded devices; and determining a time point, which is obtained by delaying a current time point by the longest idle waiting duration, as the

synchronous upgrading time point for the plurality of to-be-upgraded devices.

8. The terminal device upgrade method of claim 5, wherein, the terminal device upgrade method further comprises:

in a case where the device status information of at least one of the plurality of to-be-upgraded devices indicates that the at least one to-be-upgraded device is in a non-idle state, obtaining an idle waiting duration from the device status information of the at least one to-be-upgraded device in the non-idle state; determining a longest idle waiting duration of the at least one to-be-upgraded device; and determining a time point, which is obtained by delaying a current time point by the longest idle waiting duration, as the synchronous upgrading time point for the plurality of to-be-upgraded devices.

9. The terminal device upgrade method according to any one of claims 1 to 8, wherein, the number of the at least one upgradeable device is more than one, and before sending the to-be-upgraded software data package to each upgradeable device, the terminal device upgrade method further comprises:

determining an upgrade duration for each of the more than one upgradeable devices based on historical upgrade information; and

the sending a to-be-upgraded software data package to each of the at least one upgradeable device, comprises:

sending the to-be-upgraded software data package to a first upgradeable device of the more than one upgradeable devices, and sending a first delayed upgrade instruction to a second upgradeable device; wherein the first delayed upgrade instruction is generated based on a first upgrade duration of the first upgradeable device; and the first delayed upgrade instruction is configured to instruct the second upgradeable device to perform, after delaying for the first upgrade duration, software upgrading according to the to-be-upgraded software data package sent by the server; and in an upgrade order of the more than one upgradeable devices performing software upgrading, the second upgradeable device performs software upgrading later than the first upgradeable device.

10. The terminal device upgrade method of claim 9, wherein the terminal device upgrade method further comprises:

in a case where an upgrade pause request sent by the first upgradeable device is received during sending the to-be-upgraded software data package to the first upgradeable device, pausing sending a fragment of the to-be-upgraded software data package to the first upgradeable device, and sending a

fragment of the to-be-upgraded software data package to the second upgradeable device;

after receiving a continue sending instruction sent by the first upgradeable device, pausing sending the fragment of the to-be-upgraded software data package to the second upgradeable device, and sending a remaining fragment of the to-be-upgraded software data package to the first upgradeable device; and

after determining that the first upgradeable device completes software upgrading, sending a remaining fragment of the to-be-upgraded software data package to the second upgradeable device.

11. The terminal device upgrade method of claim 9, wherein, the more than one upgradeable devices have dependencies with each other, the first upgradeable device is a source-end upgradeable device of the more than one upgradeable devices with the dependencies; and the terminal device upgrade method further comprises:

in a case where it is determined that an actual upgrade duration of the first upgradeable device is less than the first upgrade duration and the second upgradeable device is in a sleep state, after the source-end upgradeable device upgrade completes software upgrading, sending a fragment of the to-be-upgraded software data package to a third upgradeable device of the more than one upgradeable devices; wherein, the third upgradeable device depends on the second upgradeable device;

after a target duration, sending the to-be-upgraded software data package to the second upgradeable device, and sending a second delayed upgrade instruction generated based on a second upgrade duration of the second upgradeable device to the third upgradeable device; wherein the second delayed upgrade instruction is configured to instruct the third upgradeable device to perform, after delaying for the second upgrade duration, software upgrading according to the to-be-upgraded software data package sent by the server; and the target duration is the difference between the first upgrade duration and the actual upgrade duration of the first upgradeable device.

12. The terminal device upgrade method of claim 11, wherein, for the more than one upgradeable devices with the dependencies, the terminal device upgrade method further comprises:

in a case where one of the more than one upgradeable devices finishes software upgrading early and another one of the more than one upgradeable devices depending on the early-finished one upgradeable device is in a sleep state, sending the to-be-upgraded software data package to a next upgradeable device depending on the early-finished one upgradeable device.

13. A terminal device upgrade apparatus, the apparatus being a server and comprising:

an obtaining module, configured to, in response to obtaining a software upgrade request, obtain a to-be-processed target upgrade task;

a transceiver module, configured to: send device status acquisition instruction to a plurality of to-be-upgraded devices associated with the target upgrade task, and receive device status information returned by each of the plurality of to-be-upgraded devices;

a determination module, configured to determine at least one upgradeable device from the plurality of to-be-upgraded devices based on the device status information returned by each of the plurality of to-be-upgraded devices; and

a sending module, configured to send a to-be-upgraded software data package to each of the plurality of to-be-upgraded devices, wherein the at least one upgradeable device is configured to perform software upgrading based on the to-be-upgraded software data package.

14. An electronic device, comprising:

a processor; and

a memory storing a program,

wherein the program comprises instructions; the instructions, when being executed by the processor, are configured to cause the processor to perform the method according to any one of claims 1 to 12.

15. A non-transitory computer-readable storage medium, storing computer instructions, wherein the computer instructions are configured to cause a computer to perform the method according to any one of claims 1 to 12.