



(51) International Patent Classification:

H04L 12/58 (2006.01)

(21) International Application Number:

PCT/CN2018/089557

(22) International Filing Date:

01 June 2018 (01.06.2018)

(25) Filing Language:

English

(26) Publication Language:

English

(71) Applicant: TELEFONAKTIEBOLAGET LM ERICSSON (PUBL) [SE/SE]; S-164 83 Stockholm (SE).

(71) Applicant (for SC only): ZHANG, Fengpei [CN/CN]; 5-6F, No 44-46 Jianzhong Road, Guangzhou, Guangdong 510630 (CN).

(72) Inventors: ZHANG, Fengpei; 5-6F, No 44-46 Jianzhong Road, Guangzhou, Guangdong 510630 (CN). ZHANG, Yun; ET2, No.5 Lize East street, Chaoyang District, Beijing 100102 (CN). WANG, Minyi; 5-6F, No 44-46 Jianzhong Road, Guangzhou, Guangdong 510630 (CN). TAO, Jin-grui; 5-6F, No 44-46 Jianzhong Road, Guangzhou, Guangdong 510630 (CN).

(74) Agent: CHINA SCIENCE PATENT & TRADEMARK

AGENT LTD.; Suite 4-1105, No. 87, West 3rd Ring North Rd., Haidian District, Beijing 100089 (CN).

(81) Designated States (unless otherwise indicated, for every

kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States (unless otherwise indicated, for every

kind of regional protection available): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

(54) Title: METHOD AND APPARATUS FOR PERFORMING COMMUNICATION IN INTERNET OF THINGS

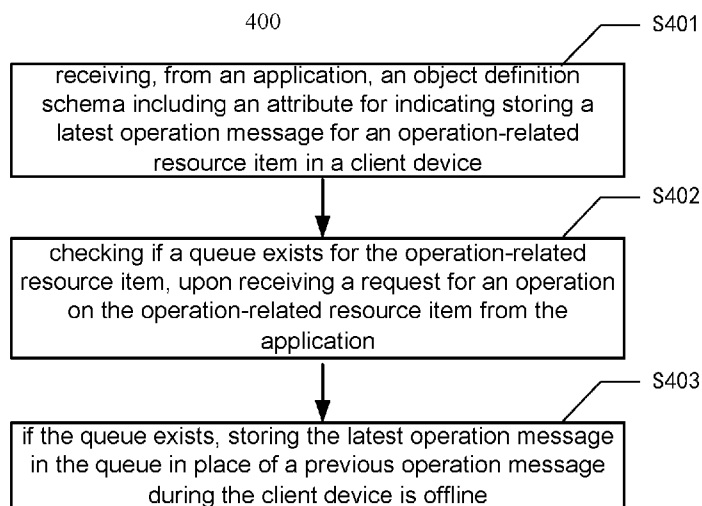


Fig. 4

(57) Abstract: The present disclosure provides a method performed in a server for communication in Internet of Things, comprising: receiving, from an application, an object definition schema including an attribute that indicates storing a latest operation message for an operation-related resource item in a client; checking if a queue exists for the operation-related resource item, upon receiving a request for an operation on the operation-related resource item from the application; and if the queue exists, storing the latest operation message in the queue in place of a previous operation message when the client is offline. The present disclosure also provides an apparatus performed in a server for communication in Internet of Things.

Published:

— *with international search report (Art. 21(3))*

METHOD AND APPARATUS FOR PERFORMING COMMUNICATION IN INTERNET OF THINGS

TECHNICAL FIELD

The present disclosure relates to the field of Internet of Things (IoT), and more particularly, to a method and apparatus for performing communication in IoT.

BACKGROUND

Open Mobile Alliance (OMA) Lightweight Machine to Machine (LWM2M) is a device management protocol designed for sensor networks and the demands of a machine-to-machine (M2M) environment. M2M environment generally implies numerous and constrained capabilities owing to low cost, small size, low processing power or limited battery life. LWM2M has been specified by a group of industry experts at the Open Mobile Alliance's Device Management Working Group and is based on protocol and security standards from the Internet Engineering Task Force (IETF). With LWM2M, OMA has responded to demand in the market for a common standard for managing lightweight and low power devices on a variety of networks necessary to realize the potential of IoT. The LWM2M protocol, designed for remote management of M2M devices and related service enablement, features a modern architectural design based on Representational State Transfer (REST), defines an extensible resource and data model and builds on an efficient secure data transfer standard called the Constrained Application Protocol (CoAP).

Referring to Figure 1, which schematically illustrates a general LWM2M architecture, the top-level applications, such as Device Management Application, M2M Web Applications, communicating with LWM2M via e.g. HyperText Transfer Protocol (HTTP) based protocols. OMA LWM2M defines the application layer communication protocol between a LWM2M server and a LWM2M client. The LWM2M protocol stack utilizes the IETF Constrained Application Protocol (CoAP)

as the underlying transfer protocol over User Datagram Protocol (UDP) and Short Messaging Service (SMS) bearers.

The CoAP is a specialized web transfer protocol designed for M2M applications in constrained nodes and constrained networks. In the present context, the term “constrained” refers to limitations in CPU, memory, and power resources that are inherent to devices which are expected to be used for IoT. CoAP defines message headers, request/response codes, message options, and retransmission mechanisms.

An LWM2M server manages LWM2M clients. An LWM2M client contains various Objects with different Resources. It executes commands sent by the LWM2M servers and then reports the results for device management and the service enablement. Several interfaces are defined between the LWM2M server and the LWM2M client, e.g., bootstrapping is used to set initial configurations on the LWM2M clients. Registration enables LWM2M clients to register with one or more LWM2M servers after bootstrapping of the LWM2M client. Object/Resource access allows the LWM2M server to perform operations on the LWM2M client, etc., and reporting allows the LWM2M client to report information to the LWM2M server in the form of Notifications.

The LWM2M defines a simple resource model where each piece of information made available by the LWM2M client is a Resource. Referring to Figure 2, which schematically illustrates a block diagram of a LWM2M client, a number of Objects may be included in a LWM2M client, and resources are logically organized into Objects. The LWM2M client may have any number of Resources, each of which belongs to an Object.

Resources and Objects have the capability to have multiple instances of the Resource or Object. A Resource is defined to have one or more operations that it supports. A Resource may consist of multiple instances called a Resource Instance as defined in the Object specification. E.g., the LWM2M server can send a “Write” operation with JavaScript Object Notation (JSON) or Tag Length Value

(TLV) format to Resource to instantiate a Resource Instance. The LWM2M client also has the capability to instantiate a Resource Instance.

Figure 3 schematically illustrates a working flowchart between the LWM2M server and the LWM2M client in the prior art. Generally, when the LWM2M client sends a registration request with a queue mode, the LWM2M server will queue the operations when the client goes offline. Once the client is online again, the LWM2M server sends the queued operations to the client.

However, in some scenarios, especially in the case that the constrained device (e.g., the LWM2M client) sleeps for a long-time period and a large number of queued operations sent from the LWM2M server are produced, this may sometimes lead to a lot of useless and redundant operations to be performed and waste network resources and the power of the LWM2M client.

SUMMARY

In view of the above, an objective of the present disclosure is to provide a method and an apparatus to enhance the current LWM2M queue mode so as to mitigate or eliminate at least one of the above problems.

Embodiments of the present disclosure provide a method and an apparatus to enhance the LWM2M queue mode by supporting store-latest messaging service. This can simplify the implementation of the LWM2M Applications and constrained devices (i.e., LWM2M clients) and improve the power consumption of constrained devices since less messages may be delivered to them.

In one aspect, there is provided a method performed in a server for communication in Internet of Things. The method comprises: receiving, from an application, an object definition schema including an attribute that indicates storing a latest operation message for an operation-related resource item in a client, checking if a queue exists for the operation-related resource item, upon receiving a request for an operation on the operation-related resource item from the application, and if the queue exists, storing the latest operation message in

the queue in place of a previous operation message when the client is offline.

With the method of the disclosure, the implementation of the LWM2M applications and the LWM2M clients that have the requirement of store-latest messaging service may be simplified; and power consumption of the LWM2M clients may be decreased.

In an embodiment, the method further comprises sending the latest operation message to the client when the client becomes online.

In an embodiment, the method further comprises sending the latest operation message to the client when receiving a registration update request from the client.

In an embodiment, the attribute that indicates storing a latest operation message is set up for the operation-related resource item or an object in the client.

In an embodiment, the method further comprises creating the queue for the operation-related resource item if the queue does not exist, when receiving a request for operation on the operation-related resource item from the application.

In an embodiment, the method further comprises receiving a request enabling a queue mode from the client.

In an embodiment, the method further comprises sending an acknowledge message to the application after storing the latest operation message in the queue.

In an embodiment, the method is adapted for communication between Lightweight Machine to Machine devices.

In another aspect, there is provided a method performed in a client for communication in Internet of Things. The method comprises receiving a latest operation message from a server when the client becomes online, wherein the latest operation message is stored in a queue created for an operation-related resource item in the client in response to receiving, by the server, from an application an object definition schema including an attribute that indicates storing a latest operation message for the operation-related resource item in the client;

and performing communication according to the latest operation message as received.

In an embodiment, the method further comprises receiving the latest operation message from the server when sending a registration update request to the server.

In an embodiment, the attribute that indicates storing a latest operation message is set up for an operation-related resource item or an object in the client.

In an embodiment, the method further comprises sending a request enabling a queue mode to the server device.

In an embodiment, the method is adapted for communication between Lightweight Machine to Machine devices.

In another aspect, there is provided a server for performing communication in Internet of Things. The server comprises: a memory adapted to store computer-executable instructions; and a processor adapted to perform the method of the embodiments of the present application, when the instructions are executed by the processor.

In an embodiment, the server is a Lightweight Machine to Machine, LWM2M, server.

In another aspect, there is provided a client for performing communication in Internet of Things. The client comprises: a memory adapted to store computer-executable instructions; and a processor adapted to perform the method of the embodiments of the present disclosure, when the instructions are executed by the processor.

In an embodiment, the client is a Lightweight Machine to Machine, LWM2M, client.

In still another aspect, there is provided a computer program for performing communication in Internet of Things, comprising computer program instructions, which, when executed on at least one processor on a server or a client, causes the server or the client performing the corresponding method according to

embodiments of the present disclosure.

In still another aspect, there is provided a computer program medium for performing communication in Internet of Things, with computer program instructions stored thereon, which, when executed on at least one processor on a server or a client, causes the server or the client performing the corresponding method according to embodiments of the present disclosure.

In still another aspect, there is provided a computer program product for performing communication in Internet of Things, comprising a computer program and a computer program medium according to embodiments of the present disclosure.

BRIEF DESCRIPTION OF THE DRAWINGS

The above and other objectives, features, and advantages of the present disclosure will be more apparent from the following description of the embodiments of the present disclosure with reference to the attached drawings, in which:

Fig. 1 schematically illustrates a general LWM2M architecture;

Fig. 2 schematically illustrates a general block diagram of a LWM2M client;

Fig. 3 schematically illustrates a working flowchart between a LWM2M server and a LWM3M client in the prior art;

Fig. 4 schematically illustrates a diagram of a method performed in a server for communication in IoT according to an embodiment of the present disclosure;

Fig. 5 schematically illustrates a diagram of a method performed in a client for communication in IoT according to an embodiment of the present disclosure;

Fig. 6 schematically illustrates an operation flowchart between a server and a client according to an embodiment of the present disclosure;

Fig. 7 schematically illustrates a block diagram of a server for performing communication in IoT according to an embodiment of the present disclosure; and

Fig. 8 schematically illustrates a block diagram of a client for performing

communication in IoT according to an embodiment of the present disclosure.

DETAILED DESCRIPTION

Hereinafter, embodiments of the present disclosure will be described with reference to the accompanying drawings. It should be understood, however, that these descriptions are only illustrative and are not intended to limit the scope of the present disclosure. In addition, in the following description, descriptions of well-known structures and techniques are omitted so as not to unnecessarily obscure the concept of the present disclosure.

All terms (including technical and scientific terms) used herein have the same meaning as commonly understood by one of ordinary skill in the art unless otherwise defined. It should be noted that terms used herein should be construed as having a meaning that is consistent with the context of the present specification and should not be interpreted in an idealized or overly formal manner.

In the context herein, the term “server” is intended to embrace any form of servers, including physical server devices or virtual servers, especially including a Lightweight Machine to Machine (LWM2M) server; the term “client” is intended to embrace any form of clients, including physical client devices or virtual clients, especially including a Lightweight Machine to Machine (LWM2M) client.

IP-based Smart Objects (IPSO) provide a common design pattern, an object model, to provide high level interoperability between Smart Object devices and connected software applications on other devices and services. IPSO Objects are defined in such a way that they do not depend on the use of CoAP, any RESTful protocol is sufficient. Nevertheless, to develop a complete and interoperable solution the Object model is based on the OMA LWM2M.

Fig.4 schematically illustrates a diagram of a method 400 performed in a server for communication in IoT according to an embodiment of the present disclosure.

The method 400 includes receiving S401 an object definition schema from an application. The object definition schema includes an attribute, e.g. “store-latest”, which is used to instruct the server to store a latest operation message for an operation-related resource item in a client.

In the context, the term “an operation-related resource item” refers to the resource item(s) for which operations/actions may be performed.

In the context, the object definition schema may be e.g. an IP-based Smart objects schema which will be explained and exemplified hereinafter.

According to an embodiment, the attribute that indicates storing a latest operation message may be set up for one or more operation-related resource items, or for one or more objects in the client.

For example and not for limitation, the application may include, e.g., various LWM2M applications; the server may include e.g. a LWM2M server, and the client may include e.g. a LWM2M client.

The method 400 further includes checking S402 if a queue exists for the operation-related resource item, upon receiving a request for an operation on the operation-related resource item from the application.

According to an embodiment, the request for an operation may be, e.g., tuning the dimmer of a light controller to 70 (the operation range of the dimmer may be, e.g., 0-100).

The method 400 further includes, if the queue exists, storing S403 the latest operation message in the queue in place of a previous operation message when the client is offline.

According to an embodiment, if the queue does not exist, the server will create a queue for the operation-related resource item when receiving a request for operating on the operation-related resource item from the application, and store the latest operation message in the queue in place of a previous operation message when the client is offline.

The previous operation message may not exist in the queue when the

queue is created initially.

According to an embodiment, the server may send the latest operation message to the client when the client becomes online again. For example, the server may send the latest operation message to the client when receiving, e.g., a registration update request from the client.

According to an embodiment, the server may receive a request for enabling a queue mode from the client. In particular, for example, the server may receive, e.g., a registration request for enabling a queue mode from the client.

Alternatively, the queue mode may be enabled via a request from the application, or may be configured by the server itself.

According to an embodiment, the server may send an acknowledge message to the application after storing the latest operation message in the queue.

For example and not for limitation, the method 400 may be applied in communications between LWM2M devices.

The embodiment of the present disclosure may simplify the implementation of the LWM2M Applications that have the requirement of store-latest messaging service since the LWM2M Applications need not always check the status of the LWM2M clients before sending the operation with the support of the LWM2M server according to the embodiment.

With the support of the LWM2M server according to the embodiment, the implementation of the LWM2M clients that have the requirement of store-latest messaging service may also be simplified since the LWM2M clients need not have logic to ignore redundant operations based on mechanisms such as time stamp or transaction ID with the support of the LWM2M server. As a result, the power consumption of the LWM2M clients may be decreased since less messages may be delivered to them.

Fig. 5 schematically illustrates a diagram of a method 500 performed in a client for communication in IoT according to an embodiment of the present

disclosure.

The method 500 includes receiving S501 by the client a latest operation message from a server when the client wakes up (i.e., online). The latest operation message may be stored in a queue created for an operation-related resource item in the client when the server receives, from an application, an object definition schema. The schema may include an attribute indicating the server to store the latest operation message received for the operation-related resource item.

The method 500 may further include performing S502 communication according to the received latest operation message.

According to an embodiment, the client may receive the latest operation message from the server when the client becomes online. For example, after a registration update request is sent to the server from the client, the latest operation message, if exists in the queue in the server, will be sent from the server to the client.

The attribute indicating the server to store the latest operation message may be set up for one or more operation-related resource items or one or more objects in the client.

According to an embodiment, the client may send a request enabling a queue mode to the server. For example, the client may send a registration request enabling a queue mode to the server.

For example and not for limitation, the method 500 may be applied in communications between Lightweight Machine to Machine devices.

According to the above embodiment, the LWM2M clients that have the requirement of store-latest messaging service may be implemented easily since the LWM2M clients need not have logic to ignore redundant operations based on mechanisms such as time stamp or transaction ID with the support of the LWM2M server. In this way, the power consumption of the LWM2M clients may be improved since less messages may be delivered to them.

Hereinafter, an LWM2M schema XSD is provided as an example as below:

This example introduces a “store-latest” attribute for a Resource, which is used to inform the LWM2M server to store the latest operation for one resource item contained in an object in the client, i.e., to instruct the LWM2M server to enable a store-latest messaging service for the operation. Additionally or alternatively, the “store-latest” attribute may be set up (or enabled) for more resource items; it may also be set up for one or more objects in the client.

As an example, the “store-latest” attribute is highlighted in black and bold style in the code as below:

```

.....
<xs:element name="Resources">
  <xs:complexType>
    <xs:sequence>
      <xs:element maxOccurs="unbounded" name="Item">
        <xs:complexType>
          <xs:sequence>
            .....
            <xs:element          name="Store-
Latest" type="xs:boolean">
            .....
            </xs:element>
            .....
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
.....

```

As an example of an object definition schema, the object definition schema may be, e.g., an IP-based Smart Objects schema. In particular, an IPSO Light Control schema is provided as an example as below:

With the extension proposed by the embodiment of the present disclosure, the dimmer operation of the light (corresponding to Resource ID "5851") will be extended with the attribute "Store-Latest" being "True". In other words, only the latest dimmer operation for the Resource ID "5851" will be stored. Please see the code highlighted in black and bold style as below:

```

.....
<Resources>
.....
  <Item ID="5851">
    <Name>Dimmer</Name>
    <Operations>RW</Operations>
    <Store-Latest>True</Store-Latest>
    <MultipleInstances>Single</MultipleInstances>
    <Mandatory>Optional</Mandatory>
    <Type>Integer</Type>
    <RangeEnumeration>0-100</RangeEnumeration>
    <Units>%</Units>
    <Description>This resource represents a light dimmer setting,
which has an Integer value between 0 and 100 as a percentage.</Description>
  </Item>
.....
</Resources>
.....

```

With the extension, the LWM2M Application can specify which resource can be handled as store-latest by the LWM2M server.

Fig. 6 schematically illustrates an operation flowchart between a server and

a client according to an embodiment of the present disclosure.

Herein, as an example, the server is illustrated as a LWM2M server, and the client is illustrated as a LWM2M client.

In [101], the LWM2M Application uploads the IPSO schema to the LWM2M server. For example, the schema may include a light control object "3331", with a "dimmer" resource "5851" whose "store-latest" attribute is enabled.

In [102], the LWM2M client may register to the LWM2M server, enabling the queue mode.

In [103], the LWM2M server may acknowledge the LWM2M client.

In [104], the LWM2M client goes offline.

In [105], the LWM2M Application may send an operation request to the LWM2M server to change the dimmer to 20.

In [106], if the LWM2M server finds out that:

1. The LWM2M client has enabled the queue mode;
2. It is an operation on 3331/5851, whose "store-latest" attribute is enabled;
3. There is no queue created for the operation-related resource item of the client (i.e., "3331/5851") before,

then, the LWM2M server may create a specific queue for this client and this type of operation for the resource item, and put this operation in the queue.

In [107], the LWM2M server may acknowledge the LWM2M Application.

In [108], the LWM2M Application may send another operation request to the LWM2M server to change the dimmer to 50.

In [109], if the LWM2M server finds out that:

1. The LWM2M client has enabled the queue mode;
2. It is an operation on 3331/5851, whose "store-latest" attribute is enabled;
3. There is already an existing queue for the operation-related resource item of the client (i.e., "3331/5851"),

Then, the LWM2M server may replace previous operation(s), if exists, in the queue with the latest one.

In [110], the LWM2M server may acknowledge the LWM2M Application.

In [111], the LWM2M client may send a registration update request to LWM2M server when the client is online (i.e., wakes up).

In [112], the LWM2M server may acknowledge the LWM2M client.

In [113], the LWM2M server may send the latest operation to the LWM2M client, which may change the dimmer to 50.

In [114], the LWM2M client may receive the operation request and acknowledge the LWM2M server.

In this example, the LWM2M client is a smart light, and the operation is light control command. According to the embodiment of the present disclosure, the smart light need not and will not receive all the historical operation commands during its sleeping period, but just the latest command when it wakes up, i.e., tunes the light dimmer to 50 accordingly.

In another example, the LWM2M client may be a smart lock, and the operation may be a remote lock/unlock command. The smart lock need not and will not receive all the historical lock/unlock commands during its sleeping period, but just the latest command when it wakes up to perform the operation accordingly.

This significantly reduces the workload of the client (e.g., the LWM2M client) and the Application side (e.g., the LWM2M Application), and reduces the power consumption of the client.

Fig. 7 schematically illustrates a block diagram of a LWM2M server 700 for performing communication in IoT according to an embodiment of the present disclosure.

The LWM2M server 700 includes a memory 701 and a processor 702. The memory 701 may store computer-executable instructions; and when the computer-executable instructions are executed by the processor, the LWM2M

server 700 is configured to receive, from an application, an object definition schema including an attribute that indicates storing a latest operation message for an operation-related resource item in a client and to check if a queue exists for the operation-related resource item, upon receiving a request for an operation on the operation-related resource item from the application. The LWM2M 700 is further configured to, if the queue exists, store the latest operation message in the queue in place of a previous operation message when the client is offline.

Alternatively, the LWM2M server may include a receiving unit configured to receive, from an application, an object definition schema including an attribute that indicates storing a latest operation message for an operation-related resource item in a client and a checking unit configured to check if a queue exists for the operation-related resource item, upon receiving a request for an operation on the operation-related resource item from the application. The LWM2M server 700 also includes a storing unit configured to, if the queue exists, store the latest operation message in the queue in place of a previous operation message when the client is offline

For example and not for limitation, the server may be, e.g., an LWM2M server.

Fig. 8 schematically illustrates a block diagram of a LWM2M client 800 for performing communication in IoT according to an embodiment of the present disclosure.

The LWM2M client 800 may include a memory 801 and a processor 802. The memory 801 may store computer-executable instructions. When the computer-executable instructions are executed by the processor, the LWM2M client 800 is configured to receive a latest operation message from a server when the client wakes up. The latest operation message is stored in a queue created for an operation-related resource item in the client in response to receiving, by the server, from an application an object definition schema including an attribute that indicates storing a latest operation message for the operation-related resource

item in the client. The LWM2M client 800 may be further configured to perform communication according to the latest operation message as received.

Alternatively, the LWM2M client may include a receiving unit configured to receive a latest operation message from a server when the client wakes up. The latest operation message is stored in a queue created for an operation-related resource item in the client in response to receiving, by the server, from an application an object definition schema including an attribute that indicates storing a latest operation message for the operation-related resource item in the client. The LWM2M client may further include a performing unit configured to perform communication according to the latest operation message as received.

For example and not for limitation, the client may be, e.g., an LWM2M client.

According to an embodiment, a computer program may be provided for performing communication in IoT. The computer program may comprise computer program instructions, which, when executed on at least one processor on a server or a client, may cause the server or the client to perform the corresponding method according to embodiments of the present disclosure.

According to an embodiment, a computer program medium may also be provided for performing communication in IoT. With computer program instructions stored thereon, the computer program medium may, when the computer program instructions are executed by at least one processor on a server or a client, cause the server or the client to perform the corresponding method according to embodiments of the present disclosure.

According to an embodiment, a computer program product may be provided for performing communication in IoT, which may include the computer program and the computer program medium as illustrated above.

In conclusion, the embodiments of the present disclosure may provide the method and apparatus supporting the Store-Latest Messaging Service for Constrained Devices. In particular, the LWM2M server may be enhanced to support the store-latest messaging service which is not yet in LWM2M

specification.

The embodiments of the present disclosure may significantly simplify the implementation of both LWM2M Applications and clients that have requirement on store-latest messaging service.

The embodiments of the present disclosure may improve the power consumption of constrained devices because with store-latest messaging service, the constrained device may receive less messages from LWM2M server.

The embodiments of the present disclosure may extend the IPSO/LWM2M schema where the LWM2M Application may define the operations that need the Store-Latest Messaging Service based on requirements.

The embodiments of the present disclosure may keep the interfaces between LWM2M Applications and clients unchanged, thus simplifying the implementation.

In the context, the “memory” herein may be implemented using any suitable data storage technology, including but not limited to semiconductor-based memory devices, magnetic memory devices and systems, optical memory devices and systems, fixed memories and removable memories.

The processor herein may include but not limited to one or more of a general purpose computer, a special purpose computer, a microprocessor, a digital signal processor DSP, and a processor based on a multi-core processor architecture.

A “computer-readable medium” as used herein shall be taken to include any medium or combination of mediums capable of storing instructions executed by a computer, a device capable of temporarily or permanently storing instructions and data, and may include, but is not limited to, random access memory (RAM), read only memory (ROM), buffer memory, flash memory, optical media, magnetic media, cache memory, other types of memory (e.g., erasable programmable read-only memory (EEPROM)) and/or any suitable combination thereof. A “computer-readable medium” can refer to a single storage apparatus or device

and/or a “cloud-based” storage system or storage network including multiple storage apparatuses or devices.

The terminology used herein is for the purpose of describing particular embodiments only and is not intended to be limiting of the disclosure. Furthermore, the terms “including”, “comprising”, or the like, as used herein, indicate the presence of stated features, steps, operations, and/or components, but does not preclude the presence or addition of one or more other features, steps, operations, or components.

Each block in the flowcharts or block diagrams in the embodiments of the present disclosure may represent a hardware module, a program segment, or a part of code, and the above-mentioned module, program segment, or part of code may include one or more of executable instructions for implementing specified logic functions.

It should also be noted that in some alternative implementations, the functions annotated in the flowcharts and blocks may also occur in a different order than that indicated in the figures. For example, two consecutively represented blocks may be executed substantially in parallel, and they may sometimes be executed in the reverse order, depending on the function involved.

It should also be noted that each block in the block diagrams or the flowcharts, and combinations of blocks in the block diagrams or the flowcharts may be implemented by a dedicated hardware-based system that performs specified functions or operations, or may be implemented by a combination of the dedicated hardware and computer instructions.

The embodiments of the present disclosure have been described above. However, these examples are for illustrative purposes only and are not intended to limit the scope of the present disclosure. Although the respective embodiments are described above respectively, this does not mean that the measures in the respective embodiments cannot be advantageously used in combination. The scope of the disclosure is defined by the appended claims and their equivalents.

Numerous substitutions and modifications may be made by those skilled in the art without departing from the scope of the present disclosure, and such substitutions and modifications should fall within the scope of the present disclosure.

CLAIMS

1. A method (400) performed in a server for communication in Internet of Things, comprising:

receiving (S401), from an application, an object definition schema including an attribute that indicates storing a latest operation message for an operation-related resource item in a client,

checking (S402) if a queue exists for the operation-related resource item, upon receiving a request for an operation on the operation-related resource item from the application, and

if the queue exists, storing (S403) the latest operation message in the queue in place of a previous operation message when the client is offline.

2. The method as claimed in claim 1, further comprising:

sending the latest operation message to the client when the client becomes online.

3. The method as claimed in claim 1 or 2, further comprising:

sending the latest operation message to the client when receiving a registration update request from the client.

4. The method as claimed in claim 1 or 2, wherein the attribute that indicates storing a latest operation message is set up for the operation-related resource item or an object in the client.

5. The method as claimed in claim 1 or 2, further comprising:

creating the queue for the operation-related resource item if the queue does not exist, when receiving a request for operation on the operation-related resource item from the application.

6. The method as claimed in claim 1 or 2, further comprising:
receiving a request enabling a queue mode from the client.
7. The method as claimed in claim 1 or 2, further comprising:
sending an acknowledge message to the application after storing the latest operation message in the queue.
8. The method as claimed in claim 1 or 2, wherein the method is adapted for communication between Lightweight Machine to Machine devices.
9. A method (500) performed in a client for communication in Internet of Things, comprising:
receiving (S501) a latest operation message from a server when the client becomes online, wherein the latest operation message is stored in a queue created for an operation-related resource item in the client in response to receiving, by the server, from an application an object definition schema including an attribute that indicates storing a latest operation message for the operation-related resource item in the client; and
performing (S502) communication according to the latest operation message.
10. The method as claimed in claim 9, further comprising:
receiving the latest operation message from the server after sending a registration update request to the server.
11. The method as claimed in claim 9 or 10, wherein the attribute that indicates storing a latest operation message is set up for an operation-related resource item or an object in the client.

12. The method as claimed in claim 9 or 10, further comprising:
sending a request enabling a queue mode to a server.

13. The method as claimed in claim 9 or 10, wherein the method is adapted for communication between Lightweight Machine to Machine devices.

14. A server (700) for performing communication in Internet of Things, comprising:

a memory (701) adapted to store computer-executable instructions; and
a processor (702) adapted to perform the method of any one of claims 1-8, when the instructions are executed by the processor.

15. The server as claimed in claim 14, wherein the server is a Lightweight Machine to Machine, LWM2M, server.

16. A client (800) for performing communication in Internet of Things, comprising:

a memory (801) adapted to store computer-executable instructions; and
a processor (802) adapted to perform the method of any one of claims 9-13, when the instructions are executed by the processor.

17. The client as claimed in claim 16, wherein the client is a Lightweight Machine to Machine, LWM2M, client.

18. A computer program for performing communication in Internet of Things, comprising computer program instructions, which, when executed on at least one processor on a server or a client, causes the server performing the method according to any one of claims 1-8 or the client performing the method according

to any one of claims 9-13 respectively.

19. A computer program medium for performing communication in Internet of Things, with computer program instructions stored thereon, which, when executed on at least one processor on a server or a client, causes the server performing the method according to any one of claims 1-8 or the client performing the method according to any one of claims 9-13 respectively.

20. A computer program product for performing communication in Internet of Things, comprising a computer program according to claim 18 and a computer program medium according to claim 19.

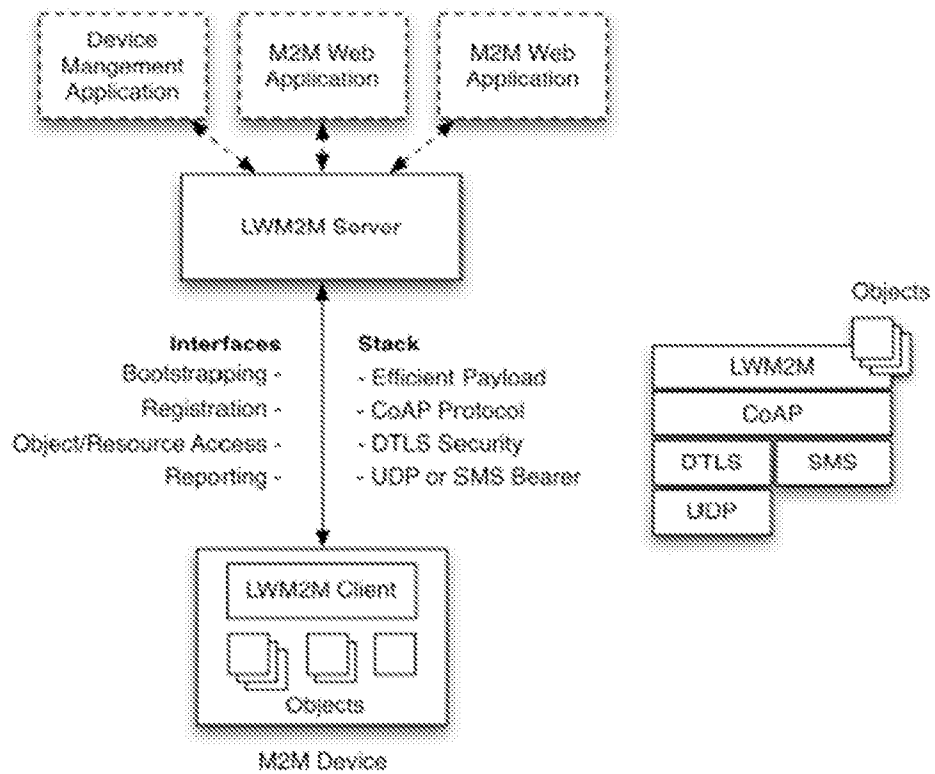


Fig. 1

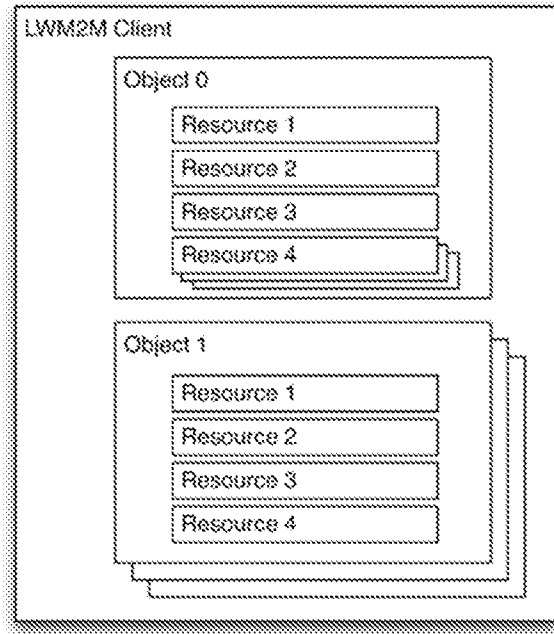


Fig.2

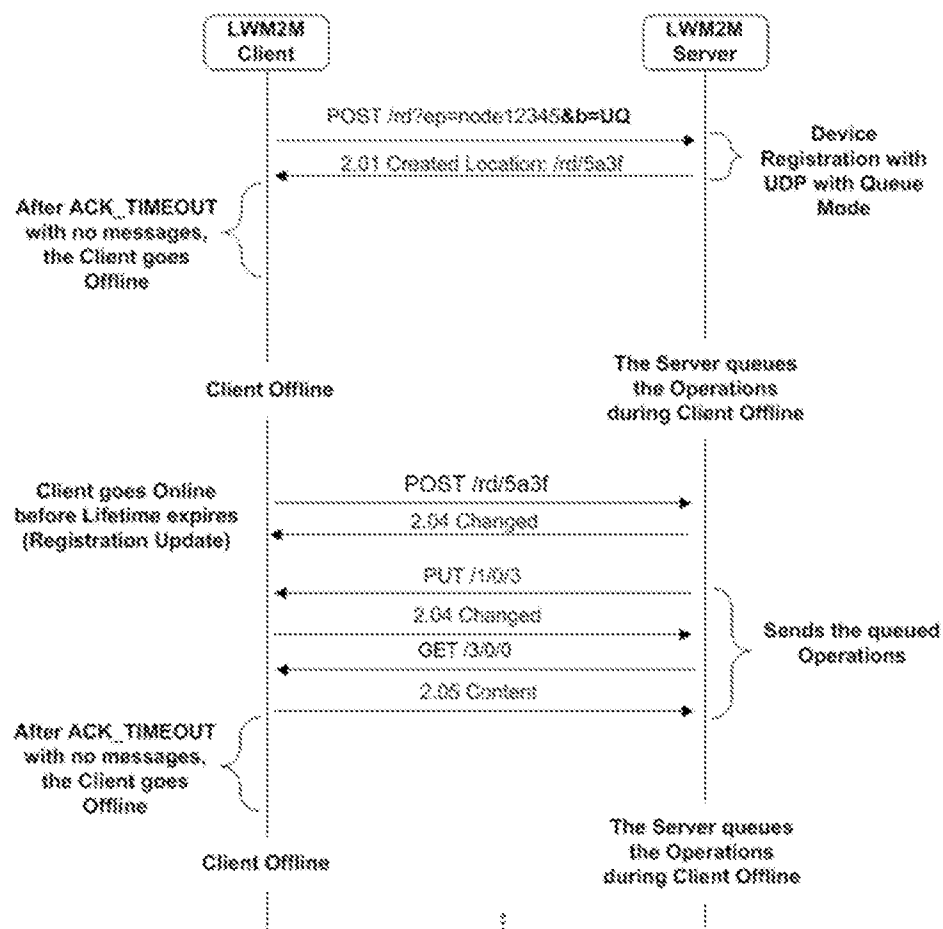
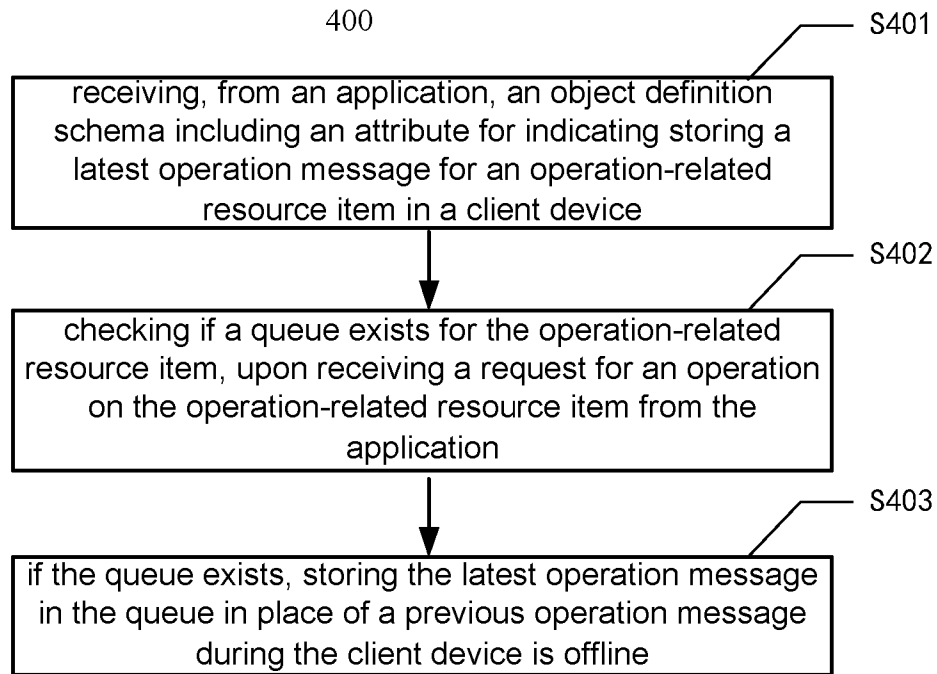
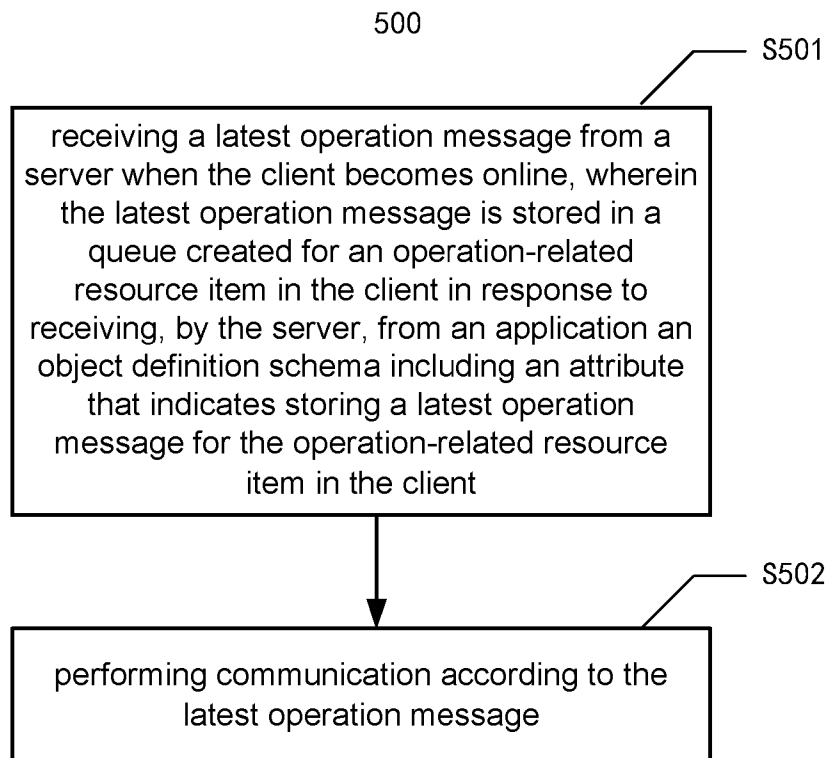


Fig. 3

**Fig. 4****Fig. 5**

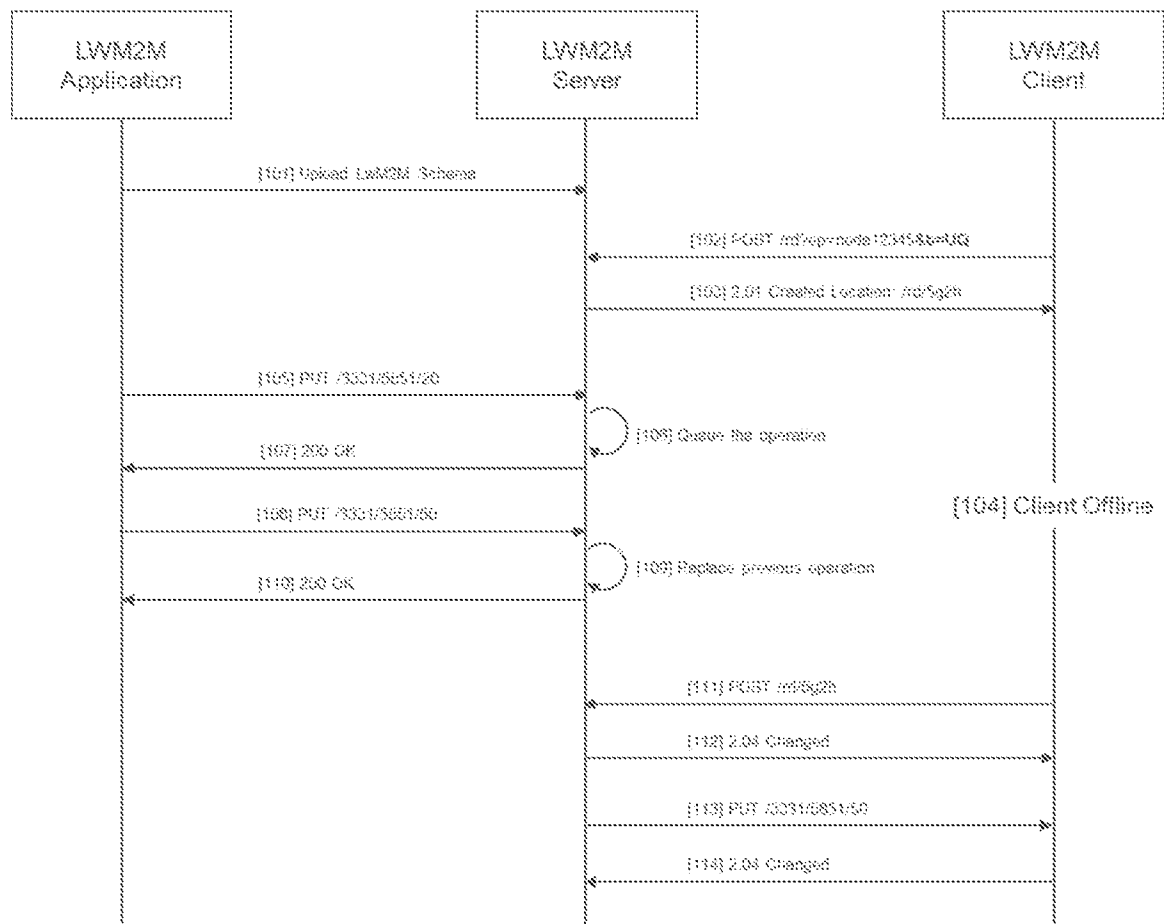


Fig. 6

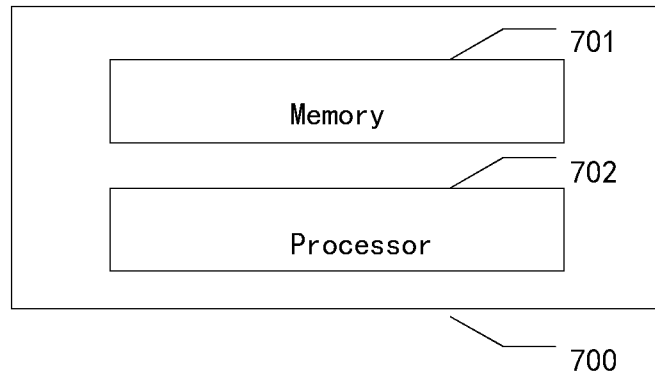


Fig. 7

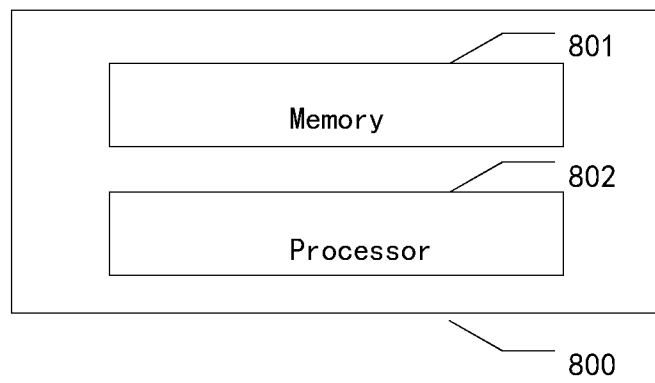


Fig. 8

INTERNATIONAL SEARCH REPORT

International application No.

PCT/CN2018/089557

A. CLASSIFICATION OF SUBJECT MATTER

H04L 12/58(2006.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

H04L H04W H04Q

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

CNPAT,CNKI,WPI,EPODOC:application, store, latest, resource, queue, offline, client, server

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	WO 2013037216 A1 (ZTE CORP.) 21 March 2013 (2013-03-21) page 1 line 15 to page 2 line 10	1-20
A	CN 104052653 A (GUANGDONG TELEPOWER COMMUNICATION CO., LTD.) 17 September 2014 (2014-09-17) the whole document	1-20
A	US 2011202909 A1 (MICROSOFT CORP.) 18 August 2011 (2011-08-18) the whole document	1-20
A	CN 105706469 A (VODAFONE IP LICENSING LTD.) 22 June 2016 (2016-06-22) the whole document	1-20

☐ Further documents are listed in the continuation of Box C.☒ See patent family annex.

* Special categories of cited documents:

“A” document defining the general state of the art which is not considered to be of particular relevance

“E” earlier application or patent but published on or after the international filing date

“L” document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

“O” document referring to an oral disclosure, use, exhibition or other means

“P” document published prior to the international filing date but later than the priority date claimed

“T” later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

“X” document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

“Y” document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

“&” document member of the same patent family

Date of the actual completion of the international search

30 January 2019

Date of mailing of the international search report

27 February 2019

Name and mailing address of the ISA/CN

National Intellectual Property Administration, PRC
6, Xitucheng Rd., Jimen Bridge, Haidian District, Beijing
100088
China

Authorized officer

LEI, Yongjun

Facsimile No. (86-10)62019451

Telephone No. 86-(10)-53961669

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.

PCT/CN2018/089557

Patent document cited in search report			Publication date (day/month/year)	Patent family member(s)			Publication date (day/month/year)
WO	2013037216	A1	21 March 2013	CN	103001984	A	27 March 2013
CN	104052653	A	17 September 2014	None			
US	2011202909	A1	18 August 2011	WO	2011100171	A2	18 August 2011
CN	105706469	A	22 June 2016	WO	2015036782	A1	19 March 2015
				GB	201415925	D0	22 October 2014
				US	2017295143	A1	12 October 2017
				US	2016234170	A1	11 August 2016
				GB	201409643	D0	16 July 2014
				GB	2520591	A	27 May 2015
				GB	2518256	A	18 March 2015
				WO	2015036778	A1	19 March 2015
				EP	3044934	A2	20 July 2016
				GB	201415003	D0	08 October 2014
				GB	201414999	D0	08 October 2014
				WO	2015036777	A1	19 March 2015
				EP	3044985	A1	20 July 2016
				US	2016232116	A1	11 August 2016
				US	2016234183	A1	11 August 2016
				GB	201409641	D0	16 July 2014
				WO	2015036776	A1	19 March 2015
				EP	3044984	A1	20 July 2016
				US	2016234182	A1	11 August 2016
				GB	2518254	A	18 March 2015
				US	2016234181	A1	11 August 2016
				GB	2519429	A	22 April 2015
				GB	201415921	D0	22 October 2014
				EP	3044983	A1	20 July 2016
				WO	2015036793	A1	19 March 2015
				WO	2015036789	A2	19 March 2015
				GB	201415918	D0	22 October 2014
				GB	201414997	D0	08 October 2014
				EP	3044928	A1	20 July 2016
				WO	2015036773	A2	19 March 2015
				US	2016234177	A1	11 August 2016
				GB	2518975	A	08 April 2015
				EP	3044982	A1	20 July 2016
				GB	2518257	A	18 March 2015
				GB	2518521	A	25 March 2015
				US	2016226847	A1	04 August 2016
				WO	2015036791	A1	19 March 2015
				GB	201409652	D0	16 July 2014
				EP	3044927	A1	20 July 2016
				US	2016234686	A1	11 August 2016
				US	2016226828	A1	04 August 2016
				WO	2015036772	A1	19 March 2015
				GB	2518255	A	18 March 2015
				WO	2015036771	A1	19 March 2015
				EP	3044976	A1	20 July 2016
				US	2016234255	A1	11 August 2016