Research on Cloud-edge Interaction Technology of Power Internet of Things Based on MQTT Protocol

Qing Liu¹,²*, Jixin Hou¹,², Yu Yan¹,², Xiaochao Wang¹,², Xiaohu Liu¹,²

¹ NARI Group Corporation (State Grid Electric Power Research Institute), Nanjing, 211106, China
² NARI Technology Co., Ltd., Nanjing 211106, China
Email: liuqing6@sgepri.sgcc.com.cn

Abstract. Under the development trend of power industry with energy Internet as the core, with the update of information technologies such as intelligent devices, mobile communication technology and cloud computing, the application of Internet of things (IOT) in power industry has developed from the concept of specific link service to the application of large-scale market. The core of the next power Internet of things construction is to build an open IOT management platform to realize the ubiquitous access of massive devices, the deep sharing of cross-professional data, business integration, and the provision of precise user services. Based on this, an industrial alliance will be formed and a power industry ecosystem with power grid enterprises as the core will be established. At present, the application of power Internet of things in China is still in the primary stage of research and application. The research on what the IOT management platform should manage and the interaction technology between the IOT management platform and edge agents is still in the stage of exploration and pilot. This paper mainly studies the cloud-edge interaction technology, including the interaction object, process and message format. Then it formulates a set of cloud-edge format specifications to meet the requirements of the platform's own management and business system message transmission, so as to provide reference for the cloud-edge interaction of power Internet of things.

Keywords: IOT management platform, cloud-edge interaction technology, edge agent, power Internet of things

1. Outline

The Internet of things connects people and people, people and things, things and things, and is a bridge for all things to transmit information [1]. The generalized Internet of things not only refers to the communication connection pipeline between all things, but also refers to the whole value creation process in which people perceive and collect the information generated by the terminal of all things. Then it can use the Internet of things network to collect, distribute and process massive data, and then display the results or feed back to the terminal through the Internet of things network and guide the terminal behavior [2].

1.1. Power Internet of Things

The power Internet of things is about all aspects of the power system, supported by advanced mobile communication technology, artificial intelligence, cloud computing and other modern information technologies. It can realize the interconnection of everything and human-computer interaction in the power network [3]. It is a comprehensive, efficient, flexible and intelligent service system. The overall
architecture of the power Internet of things can be divided into four logical levels of "cloud pipe edge terminal ", as shown in figure 1. Edge and terminal devices are located at the perception layer, the pipe corresponds to the network layer, and the cloud includes the platform layer and the application layer [4].

![Figure 1. Power Internet of things architecture.](image)

The perception layer mainly includes edge agents, intelligent terminals and intelligent applications, which can realize the interconnection and real-time monitoring of devices [5]. The network layer is supported by the advanced mobile communication network to ensure the safe, reliable, stable and efficient transmission of data [6]. The platform layer is a variety of resource business platforms built by State Grid Corporation, which cooperates with the IOT management platform, business platform and data platform to process and analyze data at the perception layer, and open and share resources. Through automatic operation and maintenance, it realizes the storage and processing of data collected by the perception layer, and provides safe and trusted channels, massive information instant messaging, device remote monitoring and other capabilities. At the same time, it can support comprehensive perception of power grid status, access management of large-scale and massive IOT terminals [7]. The application layer supports the accuracy of device management, builds a good ecology inside and outside the business. It can realize the requirements of flexible construction and rapid iteration of applications concurrently [8].

1.2. IOT Management Platform
The IOT management platform is an information system that connects the perception layer equipment with the enterprise middle office or related business systems, provides resource allocation, data aggregation and basic management functions, and supports connection management, network management, device management, user management and other functions [9]. It coordinates the IOT and perceived needs in various fields through the network layer to realize unified IOT management and terminal standardized access. Provide data support and basic component services for enterprise middle office and business applications, promote cross professional data integration and support the flexible construction of applications. It plays a connecting role in the power Internet of things [10].

1.3. Edge Agent
Edge computing is defined as an open platform close to the data source and integrating the core capabilities of network, computing, storage and application to provide localization services nearby. Its applications are initiated on the edge side to produce faster network service response and meet the basic needs of real-time business, application intelligence, security and privacy protection [11].
In the field of power Internet of things, most terminal devices lack computing and highly reliable communication ability, and cannot directly connect to the IOT management platform. It needs to complete data calculation and real-time communication by edge agent devices with edge computing ability. The edge framework program can be understood as the operating system of the edge agent device. There is one and only one edge framework program on each edge agent device, which is the infrastructure on the edge agent [12].

1.4. MQTT Protocol
Message Queuing Telemetry Transport (MQTT) is a lightweight communication protocol based on publish/subscribe mode [13]. The protocol is based on the TCP/IP protocol, uses JSON format to transmit messages, and provides real-time and reliable message services for connecting remote devices with very few codes and limited bandwidth. As an instant messaging protocol with low overhead and low bandwidth occupation, it is widely used in the Internet of things, small devices, mobile applications and so on.

2. Cloud-edge Interaction
The cloud-edge interaction of the power Internet of things mainly has the following characteristics: (1) It needs to support different types of device management across disciplines/regions. (2) It needs to support various types of application (container) management. (3) It needs to meet the transmission requirements of business data. Based on the above characteristics of the power Internet of things, this paper uses the general MQTT protocol to define the MQTT theme and message header format of cloud-edge interaction, so as to provide a unified guide for various disciplines in the power field to connect with the IOT management platform.

Cloud-edge interaction technology mainly studies the format of MQTT topic and JSON message body used for message transmission between IOT management platform and edge framework. Cloud-edge interaction scenarios mainly include:

(1) Edge and terminal device management. For the scenario that the IOT management platform manages the static information and real-time operation status information of all edge and terminal devices. The main contents include edge and terminal devices registration, access authentication, device operation status reporting, device alarm events, terminal device collection data reporting, etc.

(2) Edge app management. A scenario where the IOT management platform manages all APP programs running on each edge agent. It mainly includes the life cycle management of edge app by IOT management platform, as well as the active reporting of APP operation status and app alarm events. In practical application scenarios, applications generally run in containers, and the two are integrated. This paper is only based on app. If there are special use scenarios, they can be expanded by themselves.

(3) Edge framework management. For the scenario that the IOT management platform manages the edge framework program running on each edge agent, mainly including the remote control of the edge framework, operation status, alarm event reporting, etc.

In practical application, the message carrier can be a message in JSON format or a file, such as log file, configuration file, video file and so on. Interactive messages can be divided into three types according to functions and usage scenarios:

(1) Command and reply messages. The command message is sent to the edge app or edge framework by the application micro service or IOT management platform from top to bottom, and the reply message is bottom-up response to the command.

(2) Request and response messages. The request message is actively sent to the application micro service or IOT management platform by the edge app or edge framework from bottom to top, and the response message is the response to the request from top to bottom.

(3) Event and status messages. The event data and status data are actively sent by the edge app or edge framework to the application micro service or IOT management platform. For example, if the temperature exceeds the limit, an event data is generated, and the collected temperature value at a
certain time is status data. These messages do not need to be answered.

This chapter will study and formulate a set of cloud-edge interaction specifications for power Internet of things, and define the data interaction process and message format suitable for common scenarios of power Internet of things.

2.1. Cloud-edge Interactive Message Definition

2.1.1 Interactive Message Topic Definition. MQTT protocol message routing adopts topic mode and supports topic hierarchical definition. Universal subscription can be carried out according to the hierarchy during subscription. The use mode is flexible and simple. Each level of the theme of cloud-edge interaction is composed of lowercase letters or numbers, excluding special characters and uppercase letters. According to the characteristics of the power industry, the topic segment of MQTT is defined as follows:

1. The public field "iot", which is a required field. It is the first segment of all cloud-edge interactive topics;
2. The version field, which is a required field. It describes the version of the cloud-edge interaction topic. Currently, it defaults to V1;
3. The productId field, which is a required field. It indicates the product ID of the current edge framework. In the field of power Internet of things, different types of edge agents in the same specialty can be distinguished by topic through this field;
4. The projectIId field, which is a required field. It indicates the project ID of the current edge framework. In the field of power Internet of things, the data of edge agents of different specialty can be distinguished by topic through this field;
5. The regionId field, which is an optional field. It indicates the region identifier to which the current edge framework belongs. In the field of power Internet of things, edge agents in the same specialty, the same product and different regions can be distinguished by topic through this field;
6. The edgeld field, which is a required field. It indicates the device ID assigned by IOT management platform to the current edge device (framework);
7. The message category field, which is a required field. It indicates the scenario to which the message is applicable. In the three scenarios of edge frame, edge/terminal device and edge app, the value of this field can be "edge", "device" and "app";
8. The subcategory field of message, which is a required field. It including command/reply, request/response, event and data;
9. The segmentId field, which is an optional field, value 0, 1, 2. It can be used to extend high-frequency messages such as event and data.

Taking the interaction scenario of the edge frame as an example, the topic design is shown in table 1.

| Category | Topic | Direction | Describe |
|----------|-------|-----------|----------|
| Edge     | iot/v1/[productId]/[edgeId]/edge/command | down | Used to send the command from the platform to the edge or terminal devices |
|          | iot/v1/[productId]/[edgeId]/edge/reply | up | Used to send the command response from the edge or terminal devices to the platform |
|          | iot/v1/[productId]/[edgeId]/edge/request | up | Used to send the request from the device to the platform, such as querying the communication address of a |
2.1.2. Interactive Message Format Definition. In the field of power Internet of things, the message format of cloud-edge interaction should not only meet the management data interaction between IOT management platform and edge framework, but also support the complex business data interaction between IOT management platform and edge app. In order to standardize the data interaction between the IOT management platform and the edge agent device, and facilitate the unified data verification, this chapter defines a unified format for the message header, and uniformly restricts the use of each field of the header. The message header mainly defines the field name, field type, whether it is mandatory, and usage description, so that it can be used uniformly when defining various business scenarios, as shown in table 2.

Table 2. Message header definition.

| Field        | Type    | Required | Describe                                                                 |
|--------------|---------|----------|--------------------------------------------------------------------------|
| msg_id       | number  | Yes      | Message ID, generated by the sender                                       |
| msg_type     | string  | Yes      | Message category, customized by each business scenario                    |
| msg_name     | string  | Yes      | Message name, customized by each business scenario                       |
| object_type  | number  | Yes      | Message sending or receiving object category, enumeration type            |
| object_id    | string  | Yes      | Identification of message sending or receiving object. When object_type is 1, this field is the device ID. When object_type is 2, this field is the app name. |
| send_time    | number  | Yes      | Message sending time, UNIX timestamp, in seconds                         |
| expire       | number  | No       | Validity period, in seconds. When this field does not exist or the value is -1, it will never expire |
| body         | object  | No       | Message content in JSON format, customized by each business scenario     |
| sign         | string  | No       | Message signature                                                        |

For messages that need to be answered, the field definitions in the header of the response message are shown in table 3.

Table 3. Response message header definition.

| Field        | Type    | Required | Describe                                           |
|--------------|---------|----------|----------------------------------------------------|
| msg_id       | number  | Yes      | Same as the ID of the answered message             |
| msg_type     | string  | Yes      | Message category, customized by each business scenario |
This paper makes the following requirements for the use of JSON fields in message transmission:

(1) All JSON keys are composed of lowercase letters and underscores, and can only start with English letters.

(2) The values of message type and message name are composed of English uppercase letters, numbers and underscores, and can only start with English letters.

Various types of message objects are supported by the combination of object category and identification. Based on the scenario of power Internet of things application, the following extensibility design is carried out for the message type and the object to which the message belongs:

(1) The message type can be packet processing when there are a large number of management or business messages. For example, in the command for operation parameters, the message type can be uniformly defined as PARAM, and the message name for querying and modifying parameters are QUERY and MODIFY respectively.

(2) The object to which the message belongs can be divided into device class and app class. Devices message refers to that the message belongs to a device, such as a piece of temperature data reported by the temperature sensor, memory out of limit alarm events reported by the edge agent, etc. App message refer to that the message belongs to an app application on the device, such as the command issuance and response of meter AC sampling app, operation status events, etc.

2.2. Command Interaction
Commands are interactive messages with higher priority. In order to monitor the process of command execution, each command must have at least two response messages, as shown in figure 2. The first is that the edge framework sends an unconditional response message to the platform after receiving the command, indicating that the command has been received. The second is that after the command execution is completed, the edge framework sends a command execution result message to the platform.
In the MQTT topic classification used for the interaction between command message and response message, the subcategory field of command is "command", and the response message contains the field "reply". For example, in the edge framework management scenario, the topic of command issuance is iot/v1/{projectId}/{productId}/{edgeId}/edge/command, which is subscribed by the edge framework. Correspondingly, the topic of command response is iot/v1/{projectId}/{productId}/{edgeId}/edge/reply, which is subscribed by the IOT management platform.

The format of command message is defined in section 2.1.2. The message category and ID are generated by the IOT management platform. In the scenarios of edge framework and edge/terminal device management, the value of "object_type" is device, and the value of "object_id" is the ID of the edge agent or terminal device to which the message belongs. In the scenario of edge app management and business data interaction, the value of "object_type" is app, and the value of "object_id" is the name of the app to which the message belongs. The response message of the command must contain the code field, and the definition of the code field is shown in Table 4.

**Table 4. The code field definition.**

| Value | Meaning |
|-------|---------|
| 100   | The command has been delivered. |
| 200   | The command is executed successfully, and the execution result is included in the body field. |
| 3XX   | The command is executed failed, and the execution result is included in the body field. XX is customized by the business party. |
| 400   | The command timed out and was not executed. |

For example, the IOT management platform sends a restart command to the edge framework (the
project ID of the edge agent device is 123, the product type is 456, and the ID is 789). The command requires that the edge framework be restarted immediately. The specific cloud-edge interaction message can be defined as follows:

Interactive topic: `iot/v1/123/456/789/edge/command`

Command message:

```json
{
    "msg_id": 123,
    "msg_type": "SYS_MANAG",
    "msg_name": "REBOOT",
    "object_type": "DEVICE",
    "object_id": 789,
    "send_time": 1629355835737
}
```

2.3. Request and Response Messages

The request message is mainly aimed at the bottom-up interaction process of the message. According to the difference between the initiator and responder of the message, it can be divided into business request and response and management request and response. The business request is mainly used for the edge app to actively send business requests to the business system, and the management request is used for the edge framework to send management requests to the IOT management platform. The management request is received and returned by the platform. The business request is not processed by the platform, but directly transferred to the business system. The business system returns the result to the platform, and the platform assembles and returns the response to the device. The interaction sequence of the two request and response messages is shown in figure 3.

![Management message and business message interaction sequence diagram](image)

Figure 3. Management message and business message interaction sequence diagram.

In the MQTT topic classification used for the interaction of request and response messages, the subcategory field value of the request message is "request", and the topic used in the response message contains the field "response". For example, in the edge framework management request and response scenario, the topic used in the request message is `iot/v1/{projectId}/{productId}/{edgeId}/edge/request`, which is subscribed by the IOT management platform. The topic of the response is `iot/v1/{projectId}/{productId}/{edgeId}/edge/response`, which is subscribed by the edge
framework. The format of request and response message is defined in section 2.1.2. The response message of single request does not contain code field.

2.4. Events and Status
Event and status data is mainly used to report the event data and business data of power Internet of things devices, and there is no need to respond to the reported data. The timing relationship of data submission is shown in figure 4.

![Figure 4](image)

**Figure 4.** The timing relationship of data submission.

The subcategory field values of the request message in the MQTT topic used for event and status data reporting are event or data respectively. For example, the topic used in the scenario of frame event reporting is `iot/v1/{projectId}/{productId}/{edgeId}/edge/event`. The topic used in the scenario of frame status data reporting is `iot/v1/{projectId}/{productId}/{edgeId}/edge/data`, which is subscribed by the IOT management platform. For the message format, see 2.1.2. Section definition.

3. Pilot Application
In the distribution IOT application pilot in a city, the upper application is the distribution cloud master station, and the access device includes AC sampling, intelligent meter, shunt monitoring unit and so on, as shown in figure 5.

![Figure 5](image)

**Figure 5.** Distribution IOT pilot structure diagram.

Based on the cloud-edge interaction technology studied in this paper, the functions of edge and terminal device access, application whole process life cycle management, alarm event reporting of
edge agent device and framework, data collection and reporting of terminal devices such as AC sampling and smart meter are designed and implemented.

4. Summary
(1) This paper is based on the common MQTT protocol in the Internet of things industry to realize cloud-edge interaction in the power industry. The interaction process is clear and concise, the message format is simple and easy to and expand. At the same time, it easy to adapt and use by various professionals.

(2) As the data aggregation and forwarding node, the IOT management platform can easily realize the data distribution and diversion of different dimensions (product/project/region) on the side of the IOT management platform based on the topic scheme in this paper. For the event and status data reported frequently by the power industry, the topic can be further expanded according to the segment ID. Combined with the message partition strategy of message middleware such as Kafka, it can realize the flexible subscription and forwarding of business data, and meet the dynamic expansion of high concurrent message processing performance.

References
[1] Sun Q B, Liu J, Li Sh, et al. 2010 Internet of things: Summarize on concepts, architecture and key technology problem Journal of Beijing University of Posts and Telecommunications 33(3): 1-9.
[2] International Telecommunication Union 2005 Internet Reports 2005: The Internet of Things Geneva: ITU.
[3] Zhu N, Liang X, Zhang L F, et al. 2019 Fusion of the 5G communication and the ubiquitous electric internet of things: Application analysis and research prospects Power System Technology 43(5): 1575-1585.
[4] ITU-T Study Group 13 2012 Next Generation Networks-frameworks and Functional Models: Overview of the Internet of Things Switzerland: International Telecommunication Union, 2012.
[5] Zeng M, Wang Y Q, Li M Zh, et al. 2019 Preliminary study on the architecture and implementation plan of widespread power internet of things Smart Power 47(4): 1-7,58.
[6] Zhang N, Yang J W, Wang Y, et al. 2019 5G communication for the ubiquitous internet of things in electricity: technical principles and typical applications Proceedings of the CSEE 39(14): 4015-4024.
[7] Chen H Y, Li Zh H, Chen Y B, et al. 2020 Ubiquitous power internet of things based on 5G Power System Protection and Control 48(3): 1-8.
[8] Li X, Gong Q W, Qiao H, et al. 2010 The application of IOT in power systems Power System Protection and Control 38(22): 232-236.
[9] Zhang D, Yi K D, Lu S Y 2020 Construction of smart management platform system based on ubiquitous internet of things Digital Technology & Application 38(3): 128-129.
[10] Zhai W H, Fan B X, Liu Y 2021 Research on the construction and application of business scenario system based on power smart IoT system Electric Engineering (9): 135-136, 139.
[11] He F L, Chen J Q, Li Q H, et al. 2020 Application and development of internet of things in smart grid Power System Protection and Control 48(3): 58-69.
[12] Zhou F, Zhou H, Diao Y L 2020 Development of intelligent perception key technology in the ubiquitous internet of things in electricity Proceedings of the CSEE 40(1): 70-82.
[13] Ren H, Ma Y, Yang H B, et al. 2014 Message pushing server based on the MQTT protocol Computer Systems & Applications 23(3): 77-82.