IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 19, 2019, accepted March 28, 2019, date of publication April 1, 2019, date of current version April 15, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2908684

An IoT Unified Access Platform for Heterogeneity Sensing Devices Based on Edge Computing

LINA LAN[1,2], RUISHENG SHI[3], BAI WANG[2], AND LEI ZHANG[2]

[1]School of Network Education, Beijing University of Posts and Telecommunications, Beijing 100088, China
[2]School of Computer, Beijing University of Posts and Telecommunications, Beijing 100876, China
[3]School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing 100876, China

Corresponding author: Ruisheng Shi (shiruisheng@bupt.edu.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2016YFB0800302, in part by the National Natural Science Foundation of China under Grant 61802025, in part by the Fundamental Research Funds for the Central Universities under Grant 2018RC55, and in part by the Beijing Talents Foundation under Grant 2017000020124G062.

ABSTRACT With the rapid development of the Internet of Things (IoT), a large number of heterogeneous sensing devices are accessed in their proprietary ways to IoT applications, formed ''silos'' application mode. The growth of IoT applications has been hampered by this mode due to the barriers of resource sharing of heterogeneous sensing devices. This paper proposes an IoT access platform deployed at the network edge near the sensing devices. This architecture can enable more responsive IoT applications and provide efficient privacy protection for sensitive data. We propose a general ontology-based resource description model of IoT devices to provide a consistent view of heterogeneous sensing devices for IoT applications in the cloud. Based on this model, we propose an adaptive access method to provide unified access, control, and management of IoT devices with various intelligence levels. The access platform turns the ''silos'' application mode of IoT into a horizontal application mode, supports applications to share and reuse the resources of IoT devices. We demonstrate the efficacy of our architecture with an application case study that highlights our proposed resource description model and adaptive access method and evaluate performance improvement with experiments.

INDEX TERMS Sensing devices access platform, Internet of Things, edge computing, resource description model, adaptive access method.

I. INTRODUCTION

The Internet of things (IoT) is widely used in all walks of life, such as intelligent transportation, intelligent logistics, intelligent agriculture, intelligent power, etc. All kinds of IoT applications need sensing devices at the sensing layer to collect sensing data. At present, all kinds of IoT applications mostly access the sensing devices by their proprietary ways. The IoT application mode from the sensing layer to the upper layer is vertical ''silos'' application mode, and it is difficult for all kinds of applications to share the underlying sensing devices and sensing data [1]–[5]. This vertical application mode results in repeated deployment of sensor devices and low resource utilization, which is not conducive to the development of large-scale application of IoT. How to shield the heterogeneity between devices and make unified resource description for devices is the basis for IoT applications to share IoT sensing devices and resources. It is also a key issue to break the ''silos'' application mode.

At present, the sensing devices or terminals to be connected to IoT have a wide variety of characteristics, including diverse functions, different degrees of intelligence and different access modes [1]–[4]. For example, sensors and sensor networks that sense environmental temperature, humidity and illumination are generally accessed and managed through special gateways. Terminals of intelligent home, intelligent office and intelligent logistics systems, such as cameras and RFID tags, need special gateways or open API interfaces. Application systems need to develop special software to access the corresponding equipment and collect sensing data [6]–[8]. Some dedicated terminals with built-in SIM CARDS need a gateway to interact with the application.

Smartphones can interact directly with IoT applications. Different access modes of terminal devices bring huge

The associate editor coordinating the review of this manuscript and approving it for publication was Pietro Savazzi.

difficulties for application systems to access IoT devices. Solving unified access of heterogeneous terminals is the key to expanding the scale of IoT applications.

As a bridge between sensing devices and IoT applications, IoT gateway is an important IoT middleware and plays an important role in IoT applications [1], [4]. Some studies have proposed unified access methods for IoT [2]–[4]. These existing studies mainly focus on data transmission and forwarding, including protocol conversion and state control, etc., while there are few studies on management and control issues. The main drawback is that gateway software must be updated when new technologies or devices are developed [3], which is difficult to adapt to access of large-scale heterogeneous devices. Some proposal needs to make some changes in application software, without realizing complete application independence [5]. Therefore, providing IoT sensing devices with application-independent connectivity and allowing the automatic configuration of new devices are key technical issues that the IoT gateway is currently addressing.

There are many researches on resource description of IoT in the world and some achievements have been made [16]–[21]. It has become a trend to describe IoT resources based on ontology. The unified description of resources in existing studies is not universal or comprehensive enough and has certain limitations, and the implementation method is too complex. In this paper, we propose a general ontology-based resource description model of IoT devices to provide a consistent view of heterogeneous sensing devices for IoT applications.

The data generated by a large number of IoT sensing devices are geographically dispersed, and if all of them are centralized in cloud computing, it will exert unprecedented pressure on the network. Traditional cloud computing model is difficult to support IoT applications with high real-time requirements. Edge computing can overcome the limitations of real-time processing of IoT application which produce large amounts of data to the cloud [9], has become a new paradigm to deal with IoT and local computing needs [10], [11]. There have been some preliminary research that edge computing was applied to the IoT applications, such as smart city [12]–[14], intelligent transportation [15], etc., to obtain a better performance than the cloud computing model. Applying edge computing to the access of IoT terminal devices and the real-time processing of perception data is an effective idea to solve the problem of large-scale IoT applications.

Aiming at the resource sharing problem of the vertical application mode of IoT and the unified access problem of heterogeneous terminals, this paper deeply studies the resource description and access method of IoT, and proposes new access platform architecture of IoT based on edge computing. The IoT access platform is set between the IoT services in the cloud and mass heterogeneous terminal devices on the edge of network. The three-layer computing model of IoT services of ''IoT terminal-edge computing- cloud computing'' is established. The access platform supports unified access of heterogeneous terminals downward, and provides unified service interface for upper IoT applications upward. It supports sharing of IoT devices and resources.

The main contributions of this paper are as follows:

1) It proposes the architecture of IoT universal access platform based on edge computing. This architecture can enable more responsive IoT applications and provide efficient privacy protection for sensitive data.
2) It defines a general resource description model of the IoT devices to shield differences between heterogeneous devices and support resources sharing and reuse of IoT.
3) An adaptive access method for IoT heterogeneous terminals is proposed to support flexible and unified access, control and management of IoT heterogeneous terminals. The method mainly includes pre-configuration and mapping to adapt the various terminals connect to the access platform.
4) An application case study of Smart City Road Manhole Cover Monitoring System (SCRMCMS) is implemented based on the access platform, and the performance evaluation is carried out to verify the effectiveness of the access platform solution.

The remainder of this paper is organized as follows: In Section II, we present the related work, including ontology-based resource model for IoT, and unified access method for IoT. In Section III, we present the ''sensing device-access platform-application'' of three-layer computing model. In Section IV, we describe the overall structure and functions of the access platform of IoT, the unified description model of IoT resources based on ontology, and the adaptive access method of the IoT heterogeneous terminals. In Section V, we introduce an application case study of smart city based on the access platform and the performance evaluation of the access platform. Finally, we conclude the paper in Section VI.

II. RELATED WORK

A. ONTOLOGY-BASED RESOURCE MODEL FOR IOT

At present, there are many researches on resource description of IoT in the world and some achievements have been made. Xu *et al.* [16] proposes a data model based on semantics for medical aid system, dividing medical resources into entity resources, compound resources and state transfer resources. This data model can describe all kinds of entities and business resources in the whole medical emergency environment more perfectly. Wang *et al.* [17] proposes an ontology-based resource model of IoT, which is composed of five categories of resources, including attribute, state, control, history and privacy, and is formally described by JSON language. Seydoux *et al*. [18] proposes a core-domain modular IoT ontology proposing a vocabulary to describe connected devices and their relation with their environment, and the ontology object is instantiated in a home automation use case in this paper. Perera and Vasilakos [19] proposes a knowledge driven approach called Context Aware Sensor

Configuration Model to simplify the process of configuring IoT middleware platforms, so the data consumers can easily retrieve the data they required. They demonstrate how IoT resources can be described using semantics which can later be used to compose service work-flows. Wan *et al.* [20], [21] introduce the integration of ontology modeling with multi-agent technology to achieve dynamic resources management for IoT-based intelligent manufacturing in Industry 4.0. It can be seen that it has become a trend to describe IoT resources based on ontology. But the description of resources in existing studies often describe an application field, such as medical or industrial production field, which is not universal or comprehensive enough and has certain limitations.

B. UNIFIED ACCESS METHOD FOR IOT

In recent years, some studies have proposed unified access methods for the Internet of things. Batalla *et al.* [2] proposes a service oriented communication based on ID for unified access in IoT, connecting objects and services of IoT in a flexible way, which is applicable to scenarios where sense node location is closely related to the environmental structure in smart buildings. Zachariah *et al.* [4] put forward a general architecture that making the intelligent mobile phone as IoT gateway, the use of increasingly common BLE (Bluetooth Low Energy) consumption of the radio, IoT peripherals like sensing devices can connect to the Internet through general smartphone gateway, imagine IoT gateway global deployment will realize completely application-independent connection of the devices, hope to break the ''silos'' in the architecture. These existing studies mainly focus on data transmission and forwarding, including protocol conversion and state control, etc., while there are few studies on management and control issues. The main drawback is that gateway software must be updated when new technologies or devices are developed [3], which is difficult to adapt to access of large-scale heterogeneous devices. Pannuto *et al.* [5] propose a port-based abstraction for devices using a small wrapper layer. This device abstraction provides a consistent view of devices, and embeddable runtimes provide existing applications straightforward access to devices. This proposal needs to make some changes in application software, without supporting complete application independence. Therefore, providing IoT sensing devices with application-independent connectivity and allowing the automatic configuration of new devices are key technical issues that the IoT gateway is currently addressing.

III. THREE-LAYER COMPUTING MODEL OF ''SENSING DEVICE-ACCESS PLATFORM-APPLICATION''

The sensing layer of IoT includes a variety of sensing devices, such as temperature and humidity sensors that perceive the environment, RFID tags that support vehicle identification and item identification, etc., and cameras that support shooting and photographing, which are deployed in various places in a wide range and in a large number. These sensing devices periodically generate a large amount of real-time data, which are sent up to the application layer of IoT through the transmission layer. The IoT application system performs intelligent processing to realize intelligent services in various industries.

A large amount of perceptual data has the characteristics of real-time generation, large amount of data, strong timing and diverse data types. If raw data is directly uploaded to the application system, it will bring huge communication cost, high complexity of the application system and difficult to realize the sharing of perceptual data.

With the expansion of IoT applications and mobile applications, the traditional cloud computing model is faced with the contradiction between timely service response requirement and the problems of network bandwidth and delay. Therefore, edge computing has aroused the keen attention of researchers of IoT [9]–[11]. Edge computing is a new type of computing mode. Computing is carried out on the edge of the network to provide edge-side services, so that data can be processed timely and effectively near the source. Compared with cloud computing mode, it has obvious advantages in real-time and rapid response. Edge computing does not replace cloud computing, but complements it. Edge computing is suitable for services that need local data support, while cloud computing is suitable for services that need global data support. The architecture of edge computing is a three-layer model of ''end device - edge – cloud''. All three layers can provide resources and services for the application, and the application can choose the optimal configuration.

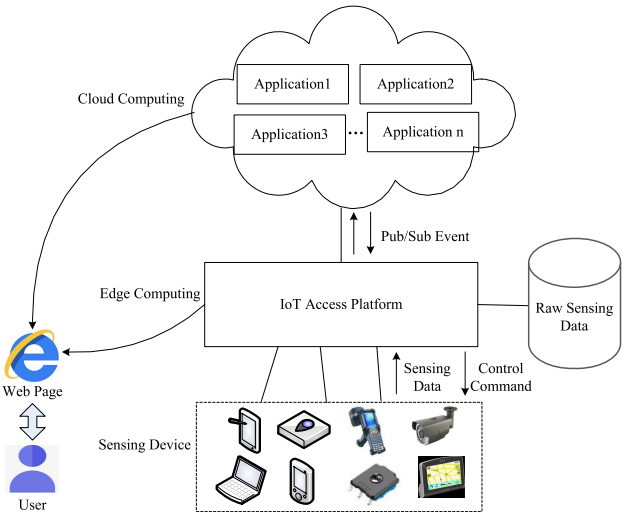


FIGURE 1. Three-layer computing model structure of ''sensing device-IoT access platform-IoT application.''

Based on edge computing, we set up IoT access platform between terminal system and IoT application service system in the cloud. The IoT access platform is deployed on the network edge to form the three-layer computing model structure of ''IoT device (terminal) - IoT access platform (edge) - IoT application (cloud computing)''. Figure 1 shows the structure.

The IoT access platform is located at the edge computing node layer, and its main functions include: supporting the access of sensing devices; supporting the collection, storage and processing of raw sensing data; uploading events of services detected in real-time; and sending down the control commands for perceptive devices by applications.

IoT application service system is located in the cloud computing layer. Through the publish/subscribe mechanism of events, IoT application system acquires events in real time through the subscription of events, and events drive the implementation of IoT services.

IoT access platform is located between IoT application service system and terminal system, serving as a bridge and a key component to change the application mode of ''silos''. The access platform of the IoT realizes the transformation of the application of the IoT from the vertical ''silos'' mode to the horizontal mode. The access platform of IoT uniformly accesses and manages the sensing devices. The application system visits the access platform of IoT, realizes the sharing and reuse of the sensing devices. The access platform spans the vertical channel from the terminal to the application, realizes the unified access and use to the terminal, and improves the utilization rate of the terminal resources.

Therefore, the IoT access platform can support unified access of a variety of heterogeneous terminals on the one hand, and the sharing of terminal resources among IoT applications on the other hand, breaking the isolation of ''silos'' applications.

In general IoT applications, sensing data will be collected in the IoT gateway, and then transmitted to the cloud for analysis and processing. The IoT access platform in this paper has the gateway and general computing capacity to collect and preprocess the original perception data, and then upload the processed data to the cloud application system for further processing.

The advantages brought by data preprocessing on the access platform of IoT before transfer to the application in the cloud are as follows:

(1) The storage and processing of original perception data run on the access platform of IoT, which is located at the edge of the network, can reduce the communication overhead of network transmission, and improve the service response speed and improve the performance of the whole system.

(2) The original data carries a variety of sensitive information, such as time and location. After preprocessing, the information that does not contain sensitive information will be released to the application in the form of events, which is conducive to the realization of privacy protection.

For example, for smart city data analyzing, the local area data analysis can be carried out in the access platform located at the edge, while the global data analysis task can be carried out in the cloud. The edge computing can realize rapid analysis and response of local data, and reduce unnecessary data transmission. Also, it is conducive to the privacy protection of users.

IV. SENSING DEVICES ACCESS PLATFORM

The IoT access platform provides unified interface services for various IoT application services, such as uploading data collected by the terminal and issuing instructions from the service system to the terminal system. It supports various terminal system accesses.

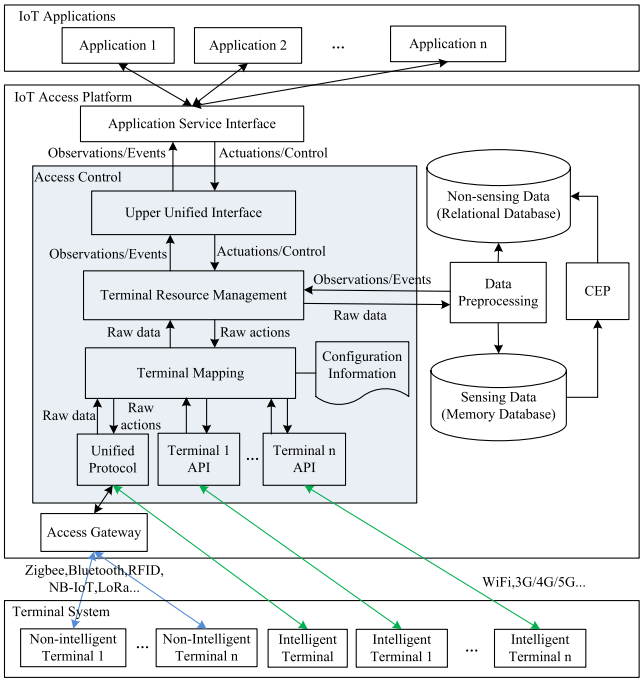


FIGURE 2. System architecture of IoT access platform.

A. PLATFORM OVERVIEW

The system structure of IoT access platform is shown in Figure 2.

In Figure 2, various IoT application systems are upon the IoT access platform. Under the IoT access platform are various terminal systems. Terminal systems include various terminals, such as non-intelligent terminal and intelligent terminal. The non-intelligent terminal refers to a terminal with a low degree of intelligence, like sensor, actuator, RFID tag and camera, etc. The intelligent terminal refers to a terminal with a high degree of intelligence, like a smart phone, network camera, etc. The terminals are connected to IoT access platform in various ways. The intelligent terminal is directly connected to the access platform, while the non-intelligent terminal is connected to the platform through the access gateway. Terminals with low intelligence are generally connected to the access gateway by wireless communication modes including Zigbee, Bluetooth, RFID, NB-IoT, LoRa, etc. The intelligent terminals connect to the access platform by WiFi or 3G/4G/5G networks. So IoT access platform supports enough kinds of communication modes to connect various sensing devices.

IoT access platform is composed of Access Gateway Module, Access Control Module, Data Preprocessing Module,

CEP (Complex Event Processing) Module and Application Service Interface Module. Among them, Access Control Module is the core module of the platform to realize the control and management of terminals.

Access Gateway Module supports the non-intelligent devices to connect to the access platform. It provides a variety of communication interfaces for the various devices. It supports the unified protocol to interact with the Unified Protocol sub-module in Access Control Module to realize the sensing devices accessing to the platform by the unified protocol.

The Data Preprocessing Module realizes the processing that needs to be done before data storage. The main functions of the module include:

1) Realize data classification storage. The sensing data stores in main memory database to acquire more fast access in order to support the real time data steaming handling of CEP. The non-sensing data stores in relational database in disk.

2) Eliminate abnormal data and wrong data from the collected data.

3) Format conversion processing, which converts different data formats of different sensory devices into a unified format for storage.

4) Privacy protection processing. Data that needs to be protected for privacy are processed for sensitive information before being stored in the database.

CEP module realizes the local complex event processing of the sensing data flow within the access platform. The raw event stream in the perceptual database is read in real time to match the pattern with the complex event rules in CEP, and the complex events are detected in real time. Complex events are written to the corresponding database and reported to the corresponding application system. CEP is not the focus of this article, so it will not be elaborated in the paper.

Application Service Interface Module is a module that provides interfaces for the upper IoT application service system. It can provide access interfaces for various service invocation modes, such as SOA services, Restful services, etc.

Between Application Service Interface Module, Access Control Module and Data Preprocessing Module are internal interfaces, which are realized through the method invocation of the class.

Access Control Module mainly deals with the control and management of terminal system, providing unified control interface and different control for different terminals. At the same time, the database needs to be read and written to obtain the necessary information to interact with the terminal and record the control history of the application. The data collected by the terminal is directed to Data Preprocessing Module for data processing. The Access Control Module includes several sub-modules, and the detailed functions are described in section C. Access Control Module is the most important module to support adaptive access method of heterogeneous terminals.

B. ONTOLOGY-BASED RESOURCE MODEL

This paper proposes a unified resource description model for IoT based on ontology, which is as follows.

1) ABSTRACT DESCRIPTION PROCESS OF IOT DEVICES

IoT sensing devices are abstracted as IoT resources, which can be accessed by IoT services. Internet resources are mainly pages, documents, services, etc., and mainly support querying and browsing. IoT resources mainly refer to the sensing data and functions of perceptive devices. They not only support identification and query, but also support state monitoring, control and other operations. Compared with Internet resources, the IoT resources are more diverse and complex.

Figure 3 shows the abstract description process of IoT devices.

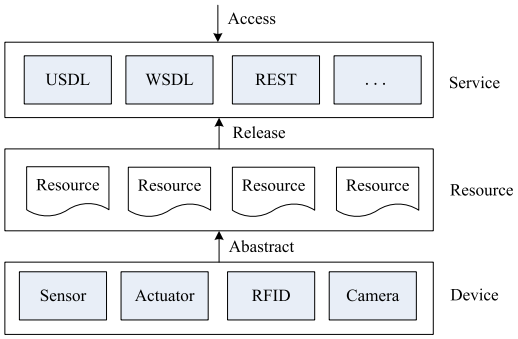


FIGURE 3. Abstract description process of IoT device.

In Figure 3, the abstract description process of IoT devices can be divided into three layers: Device Layer, Resource Layer and Service Layer. In the Device Layer, IoT devices are heterogeneous with various types, such as sensors, actuators, RFID devices and cameras, etc. These devices are abstracted into various resources at the Resource Layer, which are described in a unified form. At the Service Layer, various resources are published as Service interfaces, which can take the form of USDL (Unified Service Description Language), WSDL (Web Service Description Language), RESTful (Representational State Transfer), or any other interface described. Users or applications can access services to realize data collection of sensing devices or to execute corresponding control commands on devices.

Perceptive devices are real world entities, while resources are abstract descriptions of perceptive devices as virtual objects. Resources have two roles: (1) describing the entity, and (2) controlling the device. A service provides a resource access service for a user or application, and provides multiple access interfaces.

2) UNIFIED RESOURCE DESCRIPTION OF IOT DEVICES

Based on the above analysis, this paper proposes an ontology-based IoT resource model to achieve unified description of heterogeneous resources in IoT, and uses RDF/XML for formal description.

In this paper, the ontology model was established by using the taxonomy proposed by Perez et al. The modeling elements were classified into 5 categories, namely Classes, Relations, Functions, Instances and Rules. The ontology model RO (Resource Ontology) is obtained as in (1).

$$RO = \{C, R, F, I, RU\} \tag{1}$$

In (1):

C: represents classes, which can be subdivided into subclasses Ci;

R: represents the relationship between classes, including the following relationships:

R_p: part-of;

R_k: kind-of, inheritance relationship;

R_a: attribute-of, attribute relation;

R_i: instance-of, the relationship between instances and classes which are concepts.

F: represents the functional relationship between classes as in (2).

$$F = \{C_1 \times C_2 \times C_3 \times \ldots \times C_{n-1} \rightarrow C_n\} \tag{2}$$

I: represents the collection of instances;

RU: represents a set of rules that support inference. For example, a rule is as follows: If the indoor temperature value is over 57°, there probably will be a fire, and an alarm event should be generated.

On the basis of the ontology model, the properties and characteristics of IoT devices are analyzed, the classes in the model and the relationships between classes are abstracted, the detailed properties of each class are analyzed in depth, and the model is refined.

The IoT resources are abstracted as objects from the object-oriented point of view, and objects include basic properties and behaviors. Considering the perception and control characteristics of IoT devices, it is necessary to expand the properties of objects and add state and control. The working history of equipment is related to its historical status and historical control. The knowledge is related to domain knowledge and self-learning knowledge. Therefore, resource description should include basic attributes, state, control (or interface), historical information and knowledge information. When a physical object joins the perception and control part, it becomes the information entity of IoT. Therefore, the expanded IoT resource object description exactly reflects the integration of physical world and information space.

The definition of IoT resource object model is shown in Figure 4.

The characteristics of IoT resource object include five aspects, such as Attribute, Status, Control, History and Knowledge.

(1) Attribute: a fixed attribute of a terminal, that is, a value that does not normally change. For example, manufacturer, specification, etc. Attributes include two types, public and specific. The public attribute refers to the common attributes of all terminals, including type, model, number, manufacturer, specification, etc. The specific attribute refer to the specific attributes of a specific terminal, such as camera with focal length range, aperture range and other attributes, sensor with sensing values such as upper and lower limits. The identity needs to be defined in the public attribute to uniquely identify the terminal. Due to the variety of terminals, the type, model and serial number of terminal are used to uniquely identify each type of terminal.

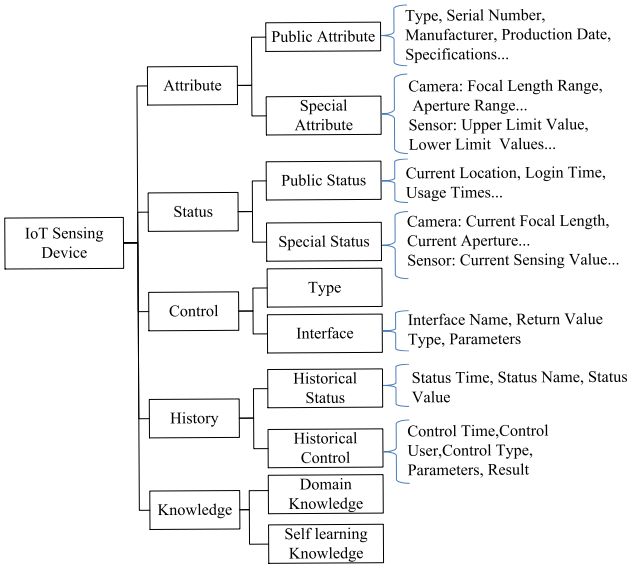


FIGURE 4. Resource object model of IoT device.

(2) Status: refers to the value of some parameters of the terminal at a certain time. For example, current location, number of uses, continuous login time, etc. Like Attribute category, Status falls into two categories, public and specific. The public status refers to the common state of various terminals, such as location, times of use, continuous login time, etc. The special status refers to the state only available at a specific terminal. For example, the camera has the current focal length value, the current aperture value and other states. The sensor has the current sensing value and other states.

(3) Control: refers to the access control interface information provided by the device. It includes the device interface type, interface description. Interface types include Web interface, API interface, etc., and interface description includes interface name, return value type, parameter information, etc.

(4) History: refers to the operation or data uploaded by the terminal in the past time. History is divided into two categories, history control (or operation) and history data (or status). Historical operation refers to the action taken by a terminal at a certain moment or the user issues a certain command to the terminal at a certain moment. For example, the sensor reports the action of sensing value at a certain moment, and the user commands the camera to turn to a certain position at a certain moment. Historical data refers to the data content reported by the terminal, such as the value of the sensing value reported by the sensor at a certain moment and the shooting content returned by the camera at a certain moment.

(5) Knowledge: includes domain knowledge and self-learning knowledge. Domain knowledge refers to the rule set of event inference in the knowledge domain of the device, which is used for event discovery. Self-learning knowledge refers to the new rules acquired by self-learning during the operation of equipment. Knowledge attributes are the basis for IoT resource objects to intelligently process perceived data and to discover important events.

IoT resource object model can realize unified resource description of various terminals by distinguishing common and special attributes of terminals.

According to the above analysis of IoT resource objects, five categories of IoT resource objects are designed as corresponding class concepts. The overall framework of resource description model based on ontology is established as shown in Figure 5.

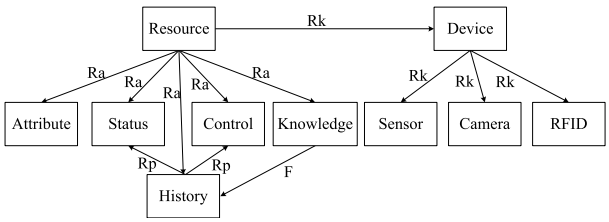


FIGURE 5. Overall architecture of IoT resource description model.

The characteristics of IoT resource classes include Attribute, Status, Control, History and Knowledge. The R_a attribute relation is between these five classes and Resource class. That is, these five classes are attributes of Resource class. The R_p inclusion relation is between the History class and the Status class and the Control class. The History class contains the Status class and the Control class. There is a functional relationship F between the Knowledge class and the History class. The event discovery rules in the Knowledge class are derived from the historical data in the History class through various logical reasoning calculations. The Device class of IoT devices can be inherited from the Resource class, and Sensor, Camera, RFID and other types of IoT devices can be inherited from the Device class. The relationship between them is R_k inheritance.

3) UNIFIED RESOURCE DESCRIPTION EXAMPLE OF IOT DEVICES

Take the temperature sensor as an example, the resource description is shown in Figure 6.

Figure 6 describes the resource of temperature sensor through four kinds of resources, including Attribute, Status, Control and History. The RDF/XML description of the temperature sensor is shown in Figure 7 below.

Since RDF/XML allows users to define data structure with their own markup language. It supports structured data and semantic expression, and describes structured data independent of applications. It realizes unified description of heterogeneous resources, and is applied to the exchange and sharing between applications.

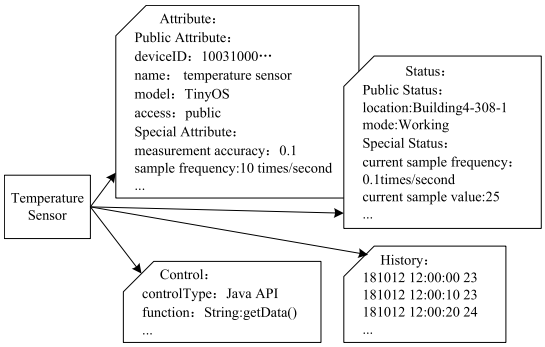


FIGURE 6. Resource description of temperature sensor.

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
         xmlns:device="http://hostserver/device#">
<rdf:Description rdf:about="http://hostserver/device/temperature sensor">
<device:attribute>
    <device:commonAttribute>
        <device:deviceID>10031000203000214120051098989823</device:deviceID>
        <device:name>temperature sensor</device:name >
      <device:type>sensor</device:type>
      <device:model>TinyOS</device:model>
        <device:access>public</device:access>
    </device:commonAttribute>
    <device:specificAttribute>
        <device:item>
            <device:name>measurement accuracy</device:name>
          <device:value>0.1</device:value >
      </device:item>
      <device:item>
          <device:name>sample frequency</device:name>
          <device:value>10</device:value>
      </device:item>
    </device:specificAttribute>
</device:attribute>
<device:control>
    <device:controlType>Java API</device:controlType>
    <device:interface>
        <device:functionID>1</device:functionID>
      <device:name>getData</device:name>
      <device:parameter></device:parameter>
      <device:result>
          <device:type>String</device:type>
        <device:comment>temperature values</device:comment>
      </device:result>
    </device:interface>
</device:control>
<device:status>
    <device:item>
        <device:stateID>1</device:stateID>
        <device:name>location</device:name>
        <device:value>Building4-308-1</device:value>
    </device:item>
    <device:item>
        <device:stateID>2</device:stateID>
        <device:name>working mode</device:name>
        <device:value>Working</device:value>
    </device:item>
    <device:item>
        <device:stateID>3</device:stateID>
        <device:name>current sample frequency</device:name>
        <device:value>0.1</device:value>
        <device:unit>times per second</device:unit>
    </device:item>
```

FIGURE 7. Temperature sensor resource description file.

```
<Resource rdf:about="#10031000203000214120051098989823_1_D101">
    <description rdf:datatype="http://www.w3.org/2001/XMLSchema#string">temperature sensor
        </description>
    <quantity_type rdf:resource="http://purl/oclc.org/NET/ssnx/qu/quantity#temperature"/>
    <unit rdf:resource="http://purl/oclc.org/NET/ssnx/qu/unit#degreeCelsius"/>
    <device_ID rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
        10031000203000214120051098989823</device_ID>
    <resource_id rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
        #10031000203000214120051098989823_1_D101</resource_id>
    <value rdf:datatype="http://www.w3.org/2001/XMLSchema#double">23</value>
    <sensing_time rdf:datatype="http://www.w3.org/2001/XMLSchema#datetime">
        2018-10-12T12:00:10</value>
</Resource>
```

FIGURE 8. Observation data resource description.

For example, if a temperature sensing data is received, the observation data can be generated according to the resource model as shown in Figure 8. The new temperature observation

data resource can provide a RESTful service to get sensing data as follows:

http://hostserver/10031000203000214120051098989823_1_D101/getData

C. Adaptive Access Method and Process

The Access Control Module includes the following modules:

1) UPPER UNIFIED INTERFACE MODULE

To provide a unified interface for different types of devices, the interface include control command, terminal serial number, control parameters. The control command represents the specific operation to be performed by the proper device. The terminal serial number is the uniform terminal serial number assigned by the access platform for the terminal. The control parameters contain the values of the parameters required for certain control commands. The output parameter of the interface is the result of an operation command and is returned as a return value.

The definition of unified interface method is shown in Figure 9 below:

```
public String termControl(String termID, String commandID, String commandValue);
/**
* termControl is the method used to operate or control the terminal.
* @param termID, the unique serial number of the terminal,String
* @param commandID, the command type to operate the terminal.
* @param commandValue, the parameter of the command, String
* @return the result of the command execution, Success or Failure.
*/
```

FIGURE 9. Unified interface method definition.

In Figure 9, the parameter termID is the unique identity of the terminal within the access platform. The parameter commandID is the command type, indicating the specific execution action of the command. The parameter commandValue is a command value that carries the parameters needed to perform the action. When the method execute, it will call the method in the interface OperationInterface.

The OperationInterface definition is shown in Figure 10:

```
public interface OperationInterface {
  public String operate (String termID, String commandValue);
  /**
  * operate is the method used to operate the terminal. It should be
implemented  in corresponding class.
  * @param termID, the unique serial number of the terminal,String
  * @param commandValue, the parameter of the command, String
  * @return the result of the command execution, Success or Failure.
  */
}
```

FIGURE 10. OperationInterface definition.

In the class that realizes this interface, concrete implementation of each operation method. All specific execution actions have implemented the above interface. According to commandID, it dynamically mapped to specific methods implementing the interface, so as to achieve the control mapping.

2) TERMINAL RESOURCE MANAGEMENT MODULE

The main function of the Terminal Resource Management Module is to transmit the control instructions issued from the Upper Unified Interface Module to the lower module, and record the control commands in the historical record. The control results returned from the lower layer are transmitted to the Upper Unified Interface Module. Send all kinds of data reported by the terminal to the Data Preprocessing Module for processing.

3) TERMINAL MAPPING MODULE

The Terminal Mapping Module mainly finds the corresponding API or protocol from the configuration file according to the terminal sequence number (including terminal type and other information) and control command, and calls the API interface or protocol of the corresponding device to complete the up unified interface and the down control of various terminals. The corresponding information such as terminal type, access mode and command interface should be configured in the configuration file. The terminal mapping module realizes adaptive access to the terminal through terminal mapping.

For example, the mapping of control commands can be done by running configuration files in the framework. In the Spring configuration file, commandID and the corresponding relationship of specific implementation class were configured. The configuration file section is shown in Figure 11 below.

```
<bean id = "sensorlogin" scope = "prototype"
  class = "com.iot.terminaloperation.sensor.SensorLogin">
</bean>
<bean id = "sensorlogout" scope = "prototype"
  class = "com.iot.terminaloperation.sensor.SensorLogout">
</bean>
<bean id = "sensorgetdata" scope = "prototype"
  class = "com.iot.terminaloperation.sensor.SensorGetData">
</bean>
```

FIGURE 11. Configuration file fragment.

The configuration described above commandID ''sensorlogin'', ''sensorlogout'' and ''sensorgetdata'', the corresponding implementation class ''SensorLogin'', ''SensorLogout'' and ''SensorGetdata''. According to commandID, the corresponding class was called to implement the command execution.

When you have a new terminal interface, you simply write a class that implements the OperationInterface interface and add the corresponding record to the configuration file to reflect the implementation class by Spring. When executing the control command, commandID was the bean id in the configuration file. And according to different commandIDs, the corresponding class was reflected for execution.

4) UNIFIED PROTOCOL MODULE

The unified protocol communication is adopted between access gateway and the unified protocol module in access platform. Based on the basic protocol design of M2M, this protocol adopts the extensible protocol stack and message structure, and has the bearing independence and strong security mechanism, which can meet the requirements of protocol flexibility, extensibility and security. Figure 12 shows the

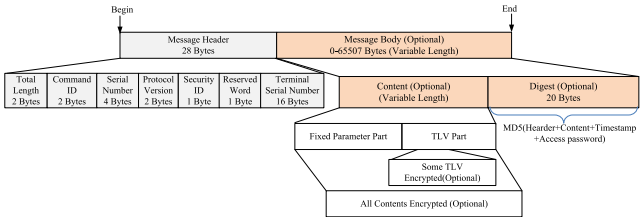


FIGURE 12. Packet structure of the unified protocol.

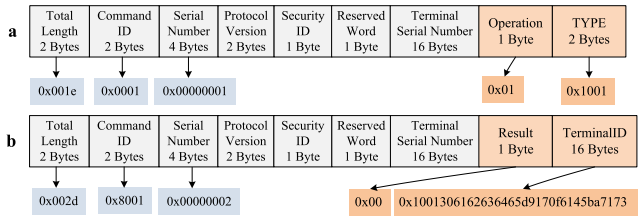


FIGURE 13. Message structures used in registration process: (a) REGISTER and (b) REGISTER_ACK.

structure of the protocol packet, which is composed of packet header and packet body.

The access platform carries out packet interaction with access gateway based on the unified protocol, which is easy to expand. Appropriate items can be added or configured in TLV (Tag Length Value) to realize unified protocol communication with new type terminals.

This unified protocol based on the unification of the IoT terminal resource description, supports message flow including the terminal access, information interaction, terminal control, terminal information management, terminal data reporting process. The message interaction process and the message are generic for various terminals. So the protocol can shield the heterogeneity of terminal in the message level. Therefore, the protocol is unified to support to all kinds of IoT terminals.

The complete process to access a new type of terminal includes the following steps: the preparatory work before connection, the terminal registration, and the normal access operation of the terminal.

(1) Preparation before connection

In the configuration file, the terminal category and interface mode are defined. Through the terminal type, the corresponding interface mode can be inquired. The access platform can use the correct interface mode to interact with the terminal to realize the adaptive access of the terminal.

(2) Terminal registration

Before the formal access, the terminal needs to register some basic information to the platform such as the type and corresponding model of the terminal. The platform assigns a unified and unique IoT terminal serial number to the terminal. The field of ''TerminalID'' is assigned by platform as shown in Figure 12(b). In the subsequent message interaction, the terminal needs to carry the unique terminal serial number for communication as shown in Figure 13(a) using the field ''Terminal Serial Number''. If encrypted communication is required, both parties of terminal and the access platform need to negotiate an encryption key at the time of registration, which is used for message authentication for the subsequent message interaction.

(3) Normal terminal access process

The normal access process of the terminal refers to the interactive process between the terminal and the platform after the successful registration. It includes Login, Data Transfer, Issue Configuration Parameters, Issue Control Commands, Log Out and other message interactions.

The unified protocol supports the above message communication process. Figure 13 shows the REGISTER and REGISTER_ACK messages. In Figure 13(b), The Result field indicates the result returned, 0x00 indicates successful registration, and the terminalID after that indicates the terminal serial number assigned by the platform. Figure 14 shows the TRANSPARENT_DATA, and TRANSPARENT_DATA_ACK messages. In Figure 14 (b), The Result field indicates the result returned, 0x00 indicates successful received. Other values indicate failure conditions, such as: 0x04 indicates platform failed to receive, CRC16 check failed, and data is invalid.

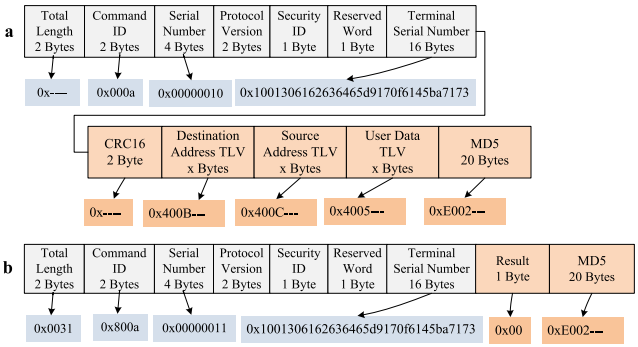


FIGURE 14. Message structures used in transfer data process: (a) TRANSPARENT_DATA and (b) TRANSPARENT_DATA_ACK.

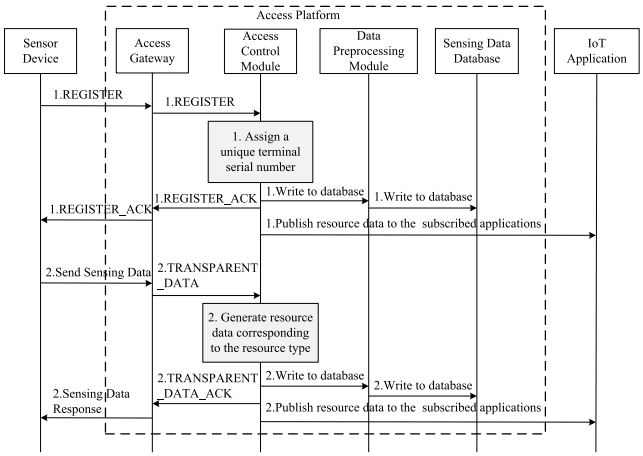


FIGURE 15. Sequence diagram of device register and data transfer process.

Figure 15 shows the two message interaction processes of registration and data transfer. It shows the message sequence

between terminal, access platform and application, and the main processing of each module.

It can be seen that various heterogeneous terminals can access the Access Platform through unified protocol, and the sensing data can be published to the application in the form of unified resources.

5) TERMINAL API MODULE

The Terminal API Module communicates with the intelligent terminal based on the API class of the terminal itself, which is mainly applicable to the intelligent terminal. Intelligent terminals can be divided into two access modes of unified protocol access and API access. Some intelligent terminals can directly access to the access platform through the unified protocol and directly interact with the access platform. Some intelligent terminals have open API provided by manufacturers, and access platform can use terminal API to develop interfaces and realize the interaction with intelligent terminals.

It can be seen that the above two modules, the Unified Protocol and Terminal API, support access of various types of terminals. Non-intelligent terminals or low-intelligent terminals need gateway to interact with the access platform by means of unified protocol. So the gateway has the corresponding protocol adaptor to interact with the new access terminal to support sensing data upload or control command download. Intelligent terminal can be accessed to platform through unified protocol or through terminal API module.

Above all, in IoT access platform, to realize the adaptive terminal access mechanism mainly includes the several steps as follows:

(1) To distinguish terminals into the intelligent terminal and non-intelligent terminal. The terminal interfaces are defined in the configuration file according to the terminal type. The query of the corresponding interface mode is defined also. So the access platform can use the right way to interact with a terminal to realize the adaptive access.

(2) The access platform realizes the required access mode in advance, which is configured in the configuration file.

(3) When terminal is connecting to the access platform, the IoT access platform can identify the terminal type according to the unique terminal serial number, and use terminal mapping protocol or API to achieve the access of heterogeneous terminals through adaptive protocol or API.

The advantage of the adaptive access method provided by the access platform is as follows:

(1) It provides application-independent connectivity of IoT sensing devices for IoT applications. It decouples the applications with sensing devices. The applications can access various heterogeneous sensing devices using the unified interface.

(2) It provides almost completely automatic configuration of new devices accessing to platform. It supports unified access, control and management of IoT devices with various intelligence levels.

V. CASE STUDY AND EXPERIMENTAL EVALUATION

A. SMART CITY APPLICATION BAESD ON THE ACCEESS PLATFORM

Based on the IoT access platform, this paper implements an IoT application system - Smart City Road Manhole Cover Monitoring System (SCRMCMS). The system supports the access of Inclination Angle Sensor and Speed Sensor, so as to support the monitoring of abnormal events such as manhole cover tilt and movement events. It supports the access of sound alarm equipment, such as buzzer, so as to support the sound alarm in case of abnormal events. It supports camera access, so as to support real-time video image monitoring. The SCRMCMS system realizes the comprehensive monitoring of road manhole covers by connecting the above sensing devices through the access platform.

Figure 16 shows the SCRMCMS system.

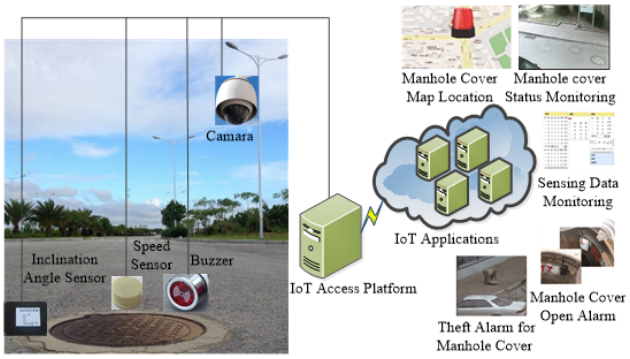


FIGURE 16. Smart city road manhole cover monitoring system.

In Figure 16, firstly, Cameras are deployed above the manhole covers of key sections in the city road, and installed on lamp posts on both sides of the road to take photos and video of the manhole covers. Then, Inclination sensor, Speed sensor and Buzzer are installed at the bottom of the cover. The buzzer is used for sound alarm. Finally, connect the above devices to the IoT Access Platform. The collected sensory data are stored into the access platform. The access platform provides data release function, and provides application access in the form of RESTful services. The SCRMCMS system can realize comprehensive and real-time monitoring of road covers, efficiently realize local data storage and data release, and support IoT applications in the cloud to share perceived resources and data.

During the system operation, the camera and sensor upload the camera data and manhole cover location data to the access platform in real time, and then the access platform transmits the data to the manhole cover monitoring system. The application system analyzes the perception data in real time and finds out business events, such as manhole cover tilt and manhole cover movement, and sends out alarm in real time, such as controlling the buzzer to send out sound alarm. Manhole cover abnormal status alarm information is displayed on the visual interface of the monitoring system to remind the management person. At the same time, the alarm

information will be informed to the video security system, and the remote camera will be used to track the occurrence of the event and locate the manhole cover.

In order to shield the heterogeneity of each sensing device in the interface and data format, and realize the unified management of the system, this experiment carried out a unified resource description of each device on the basis of the resource description model of IoT devices proposed. Based on this, the system shields the heterogeneity between devices in the bottom layer, realizes the information and data sharing between devices, and lays a foundation for the final realization of rich business functions and applications.

Different types of devices have different kinds of properties and functions, which are reflected in the description model. Detailed analysis of equipment resources is shown in Table 1.

TABLE 1. Class resource of devices.

Device name	Attribute	Status	Control	History	Knowledge
Inclination Sensor	Y	Y	Y/N	Y	Y
Speed Sensor	Y	Y	Y/N	Y	Y
Buzzer	Y	Y	Y	Y	Y
Camera	Y	Y	Y	Y	Y

The Inclination Angle Sensor and Speed Sensor involved in this experiment is mainly equipped with data acquisition function, which can be viewed through Status resources. Some advanced sensor device can achieve the control command to configure the data acquisition cycle or sample frequency, so it can has Control resource. Some low-grade sensor cannot be controlled, it has not Control resource. At the same time, the device itself supports attributes, historical information and knowledge, so it has Attribute(Y), Status(Y), Control(Y/N), History(Y) and Knowledge(Y) resources as shown in Table 1.

While the Buzzer can issue sound alarm through control, therefore, Buzzer has Attribute(Y), Status(Y), Control(Y), History(Y) and Knowledge(Y) resources as shown in Table 1.

Camera equipment has camera shooting function to acquire video data, as well as photograph and other control functions. Therefore, it has Attribute(Y), Status(Y), Control(Y), History(Y) and Knowledge(Y) resources as shown in Table 1.

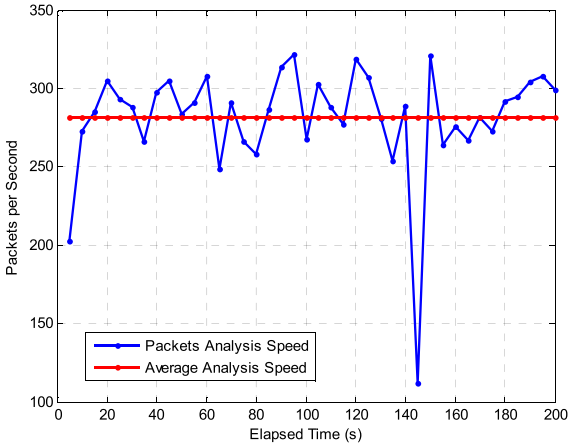
In this case study, the camera is accessed through the API mode, and the sensor and buzzer are accessed through the unified protocol of the gateway. It verifies the adaptability access of the heterogeneous terminals to the access platform.

Through experimental verification, the IoT model proposed in this paper can better describe various IoT sensing devices, and the adaptive access method proposed in this paper can better support access of various IoT devices, therefore the proposal has good application value and prospect.

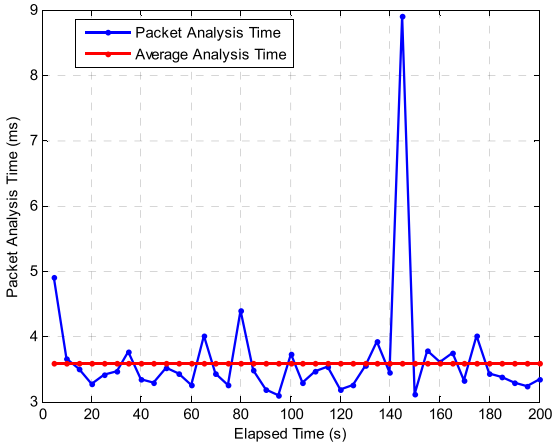
B. PERFORMANCE EVALUATION

A large number of sensing devices need to connect to the IoT access platform. A large amount of sensing data is generated in real time and uploaded to the IoT access platform through unified protocol. The processing capability of data packets of the IoT access platform is very important, which directly affects the performance of the IoT access platform. So some experiments are necessary to evaluate the protocol analysis performance of the IoT access platform.

In order to evaluate the performance of the IoT access platform, some experiments have been done. The experiments were carried out on an Intel Xeon(R) CPU E5-2682 v4 2.50 GHz processor, with 2 GB of RAM running on Windows Server 2012 R2 Datacenter x64.



(a)



(b)

FIGURE 17. Performance of protocol processing of Access Platform. (a) Packets per second. (b) Packet analysis time.

We simulate 100 data senders to send real protocol packets independently. The packets include Login, Data Transfer, Issue Configuration Parameters, Log Out and other messages. Most of the packets are Data Transfer messages. The IoT access platform analyzes these packets according to the real and completed analysis process. Figure 17 (a) shows that

the max packets-per-second is 322, the min of it is 112, and the average analysis speed is 282 packets per second. Figure 17 (b) shows that the max packet analysis time is 8.912 milliseconds, the min of it is 3.099 milliseconds, and the average packet analysis time is 3.594 milliseconds. In the experiment, the average length of messages is 228 Bytes. The results show that our platform is competent for large-scale and time-critical IoT applications.

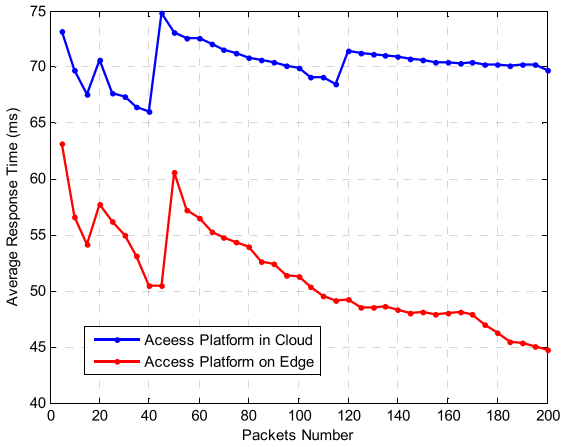


FIGURE 18. Performance of response time to terminals of Access Platform.

We do another experiment to make a comparison between the access platform running on the edge of the network (in the same LAN with the testing computer of data sender) and access platform running in the cloud (Ali Cloud). We compared the response time to the terminal when terminal sending different number of packets from 5 to 200 packets to the access platform in both cases, as shown in the Figure 18. The response time of access platform running at the edge is less than that of access platform running in the cloud. The average response time of the access platform processing 200 packets on edge is about 44.795 milliseconds, and the average response time is about 69.78 milliseconds when the access platform running in the cloud. So the response time to terminals of the access platform on the edge is about 25 milliseconds faster than the access platform in the cloud. The results show that the architecture of our platform based on edge computing can achieve better performance of response time to terminals.

VI. CONCLUSIONS

This paper proposes a unified access platform of IoT based on edge computing which is oriented to the shortcoming of ''silos'' application mode. It enables to access large-scale heterogeneous devices and expose resource capabilities as unified standard service interfaces. The general ontology-based resource description model of IoT devices provides a consistent view of heterogeneous sensing devices for IoT applications in the cloud. It decouples the upper applications with the lower sensing device. The access platform promotes creating device-agnostic applications, and supports applications to share and reuse the resources of IoT devices.

Moreover, this paper proposes an adaptive unified access method for heterogeneous terminals. It provides unified control interface for the upper application. The control instruction can be distinguished only according to the different parameters passed in, so as to facilitate the upper application to invoke the control command of the terminal. The appropriate communication mode is adopted to complete the interaction with the terminal adaptively by using pre-configuration and mapping method. The corresponding protocol or API can be used adaptively to connect various devices.

Finally, we demonstrate the efficacy of our architecture with an application case study SCRMCMS. The system highlights our proposed resource description model and adaptive access method. We evaluate the performance of the access platform, and the results show that the platform is competent for large-scale and time-critical IoT applications, and the access platform deployed at the edge can achieve better performance of response time to terminals than in the cloud.

The access platform turns the ''silos'' application mode of IoT into a horizontal application mode. This will have a positive impact on the growth of IoT applications.

The future work will focus on the IoT cooperative service provision based on the access platform.

REFERENCES

[1] A. H. Ngu, M. Gutierrez, V. Metsis, S. Nepal, and Q. Z. Sheng, ''IoT middleware: A survey on issues and enabling technologies,'' *IEEE Internet Things J.*, vol. 4, no. 1, pp. 1–20, Feb. 2017.

[2] J. M. Batalla, M. Gajewski, W. Latoszek, C. X. Mavromoustakis, and G. Mastorakis, ''ID-based service-oriented communications for unified access to IoT,'' *Comput. Elect. Eng.*, vol. 52, pp. 98–113, May 2016.

[3] O. Bello, S. Zeadally, and M. Badra, ''Network layer inter-operation of device-to-device communication technologies in Internet of Things (IoT),'' *Ad Hoc Netw. J.*, vol. 57, pp. 52–62, Mar. 2017.

[4] T. Zachariah, N. Klugman, B. Campbell, J. Adkins, N. Jackson, and P. Dutta, ''The Internet of Things has a gateway problem,'' in *Proc. ACM 16th Int. Workshop Mobile Comput. Syst. Appl. (HotMobile)*, Santa Fe, NM, USA, 2015, pp. 27–32.

[5] P. Pannuto, W. Wang, P. Dutta, and B. Campbell, ''A modular and adaptive architecture for building applications with connected devices,'' in *Proc. IEEE 1st Int. Conf. Ind. Internet (ICII)*, Seattle, WA, USA, Oct. 2018, pp. 1–12.

[6] J. Domaszewicz *et al.*, ''Soft actuation: Smart home and office with human-in-the-loop,'' *IEEE Pers. Commun.*, vol. 15, no. 1, pp. 48–56, Jan./Mar. 2016.

[7] A. C. Olivieri, G. Rizzo, and F. Morard, ''A publish-subscribe approach to IoT integration: The smart office use case,'' in *Proc. IEEE 29th Int. Conf. Adv. Inf. Netw. Appl. Workshops*, Gwangiu, South Korea, Mar. 2015, pp. 644–651.

[8] S. Alletto *et al.*, ''An indoor location-aware system for an IoT-based smart museum,'' *IEEE Internet Things J.*, vol. 3, no. 2, pp. 244–253, Apr. 2016.

[9] A. V. Dastjerdi and R. Buyya, ''Fog computing: Helping the Internet of Things realize its potential,'' *Computer*, vol. 49, no. 8, pp. 112–116, Aug. 2016.

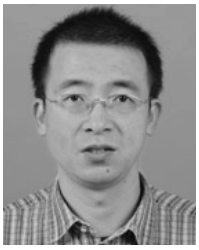
[10] W. Yu *et al.*, ''A survey on the edge computing for the Internet of Things,'' *IEEE Access*, vol. 6, pp. 6900–6919, 2018.

[11] A. Munir, P. Kansakar, and S. U. Khan, ''IFCIoT: Integrated fog cloud IoT: A novel architectural paradigm for the future Internet of Things,'' *IEEE Consum. Electron. Mag.*, vol. 6, no. 3, pp. 74–82, Jul. 2017.

[12] Q. Zhang, Z. Yu, W. Shi, and H. Zhong, ''Demo abstract: EVAPS: Edge video analysis for public safety,'' in *Proc. IEEE/ACM Symp. Edge Comput. (SEC)*, Washington, DC, USA, Oct. 2016, pp. 121–122.

[13] N. Chen, Y. Chen, S. Song, C.-T. Huang, and X. Ye, "Poster abstract: Smart urban surveillance using fog computing," in *Proc. IEEE/ACM Symp. Edge Comput. (SEC)*, Washington, DC, USA, Oct. 2016, pp. 95–96.
[14] J. He, J. Wei, K. Chen, Z. Tang, Y. Zhou, and Y. Zhang, "Multitier fog computing with large-scale iot data analytics for smart cities," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 677–686, Apr. 2018.
[15] J. Lin, W. Yu, X. Yang, Q. Yang, X. Fu, and W. Zhao, "A real-time en-route route guidance decision scheme for transportation-based cyberphysical systems," *IEEE Trans. Veh. Technol.*, vol. 66, no. 3, pp. 2551–2566, Mar. 2017.
[16] B. Xu, L. D. Xu, H. Cai, C. Xie, J. Hu, and F. Bu, "Ubiquitous data accessing method in IoT-based information system for emergency medical services," *IEEE Trans. Ind. Informat.*, vol. 10, no. 2, pp. 1578–1586, May 2014.
[17] S. Wang, Y. Hou, F. Gao, and S. Ma, "Ontology-based resource description model for Internet of Things," in *Proc. IEEE Int. Conf. Cyber-Enabled Distrib. Comput. Knowl. Discovery (CyberC)*, Oct. 2016, pp. 105–108.
[18] N. Seydoux, K. Drira, N. Hernandez, and T. Monteil, "IoT-O, a core-domain IoT ontology to represent connected devices networks," in *Proc. Eur. Knowl. Acquisition Workshop*. Cham, Switzerland: Springer, 2016, pp. 561–576.
[19] C. Perera and A. V. Vasilakos, "A knowledge-based resource discovery for Internet of Things," *Knowl.-Based Syst.*, vol. 109, pp. 122–136, Oct. 2016.
[20] J. Wan *et al.*, "Toward dynamic resources management for IoT-based manufacturing," *IEEE Commun. Mag.*, vol. 56, no. 2, pp. 52–59, Feb. 2018.
[21] J. Wan, B. Yin, D. Li, A. Celesti, F. Tao, and Q. Hua, "An ontology-based resource reconfiguration method for manufacturing cyber-physical systems," *IEEE/ASME Trans. Mechatron.*, vol. 23, no. 6, pp. 2537–2546, Dec. 2018.

LINA LAN is currently an Associate Professor with the School of Network Education, Beijing University of Posts and Telecommunications, Beijing, China. Her current research interests include the Internet of Things, intelligent information processing, and software architecture.



RUISHENG SHI received the Ph.D. degree in information and communication engineering from the Beijing University of Posts and Telecommunications, China, in 2013, where he is currently an Associate Professor with the School of Cyberspace Security. His current interests include the Internet of Things, service computing, publish/subscribe system, and information security.

BAI WANG is currently a Professor with the School of Computer Science, Beijing University of Posts and Telecommunications, Beijing, China. Her current interests include the Internet of Things, communication software, distributed computing technology, and intelligent information processing.

LEI ZHANG is currently a Professor with the School of Computer Science, Beijing University of Posts and Telecommunications, Beijing, China. His current interests include the Internet of Things, communication software, network service, and intelligent information processing.