International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

J.1612

(01/2022)

SERIES J: CABLE NETWORKS AND TRANSMISSION OF TELEVISION, SOUND PROGRAMME AND OTHER MULTIMEDIA SIGNALS

Artificial intelligence (AI) assisted cable networks – Requirements for the set-top box

The architecture for a smart home gateway

Recommendation ITU-T J.1612

Recommendation ITU-T J.1612

The architecture for a smart home gateway

Summary

Smart home is a kind of home automation system in which a wide range of IoT devices in a home cooperate to provide intelligent controlling and monitoring functions for home users. Smart home gateway connects various smart home devices, provides hardware interfaces of various smart home communication protocols, runs communication protocols, performs protocol conversion and bridging, realizes the interaction between user control terminal and Cloud server.

Recommendation ITU-T J.1612 aims to define the architecture for a smart home gateway (SHGW) which addresses the functional requirements found in Recommendation ITU-T J.1611. The Recommendation consists of concepts of a virtual device model, dynamic device profile and other important software modules. With the introduction of these important modules, the architecture can dynamically support existing smart home devices and the devices in the future.

History

Edition	Recommendation	Approval	Study Group	Unique ID*
1.0	ITU-T J.1612	2022-01-13	9	11.1002/1000/14845

Keywords

Smart home, smart home gateway, universal control platform, virtual device model.

* To access the Recommendation, type the URL http://handle.itu.int/ in the address field of your web browser, followed by the Recommendation's unique ID. For example, http://handle.itu.int/11.1002/1000/11830-en.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications, information and communication technologies (ICTs). The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure, e.g., interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents/software copyrights, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the appropriate ITU-T databases available via the ITU-T website at http://www.itu.int/ITU-T/ipr/.

Table of Contents

		Page
1	Scope	1
2	References	1
3	Definitions	1
3.1	Terms defined elsewhere	1
3.2	Terms defined in this Recommendation	1
4	Abbreviations and acronyms	2
5	Conventions	2
6	The overview of smart home service	2
7	Gateway software architecture	4
7.1	A universal control platform	5
7.2	Communication interfaces and connectivity driver	6
7.3	Virtual device model	6
7.4	Device profile and profile server	6
7.5	RESTful API	8
7.6	Gateway control agent	8
7.7	Device management	8
7.8	Cloud interface	9
Bibliography		10

Recommendation ITU-T J.1612

The architecture for a smart home gateway

1 Scope

This Recommendation provides the architecture for a smart home gateway (SHGW) that addresses the functional requirements found in [ITU-T J.1611]. The scope of this Recommendation consists of concepts of a virtual device model, dynamic device profile and other important software modules. With the introduction of these important modules, the architecture can dynamically support existing smart home devices and the devices in the future.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T J.1611] Recommendation ITU-T J.1611 (2021), *Functional requirements for a smart home gateway*.

[IETF RFC 7252] IETF RFC 7252 (2014), *The Constrained Application Protocol (CoAP)*.

[IETF RFC 8446] IETF RFC 8446 (2018), *The Transport Layer Security (TLS) Protocol Version 1.3*.

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 device [b-ITU-T Y.4000]: With regard to the Internet of things, this is a piece of equipment with the mandatory capabilities of communication and the optional capabilities of sensing, actuation, data capture, data storage and data processing.

3.1.2 gateway [b-ITU-T Y.4101]: A unit in the Internet of things that interconnects the devices with the communication networks. It performs the necessary translation between the protocols used in the communication networks and those used by devices.

3.1.3 internet of things (IoT) [b-ITU-T Y.4000]: A global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving interoperable information and communication technologies.

3.1.4 smart home gateway [ITU-T J.1611]: A device or equipment which connects various home IoT devices and performs as a bridge between a smart home cloud and a smart home device.

3.2 Terms defined in this Recommendation

This Recommendation defines the following terms:

None.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

BLE Bluetooth Low Energy

CoAP Constrained Application Protocol

DM Device Management

HTTP Hypertext Transfer Protocol

IoT Internet of Things

JSON JavaScript Object Notation

NoSQL Not only SQL

REST Representational State Transfer

SHGW Smart Home Gateway

SOAP Simple Object Access Protocol

XML Extensible Markup Language

5 Conventions

In this Recommendation:

The keywords **"is required to"** indicate a requirement that must be strictly followed and from which no deviation is permitted if conformance to this document is to be claimed.

The keywords **"is recommended"** indicate a requirement that is recommended but which is not absolutely required. Thus, this requirement need not be present to claim conformance.

The keywords **"is prohibited from"** indicate a requirement that must be strictly followed and from which no deviation is permitted if conformance to this Recommendation is to be claimed.

The keywords **"is optionally required"** indicate an optional requirement that is permissible, without implying any sense of being recommended. This term is not intended to imply that the vendor's implementation must provide the option and the feature can be optionally enabled by the network operator/service provider. Rather, it means the vendor may optionally provide the feature and still claim conformance with the specification.

The keywords **"mandatorily"** indicate a mandatory requirement which is recommended in any sense. This term is intended to imply that the vendor must implement the option and the feature. Otherwise, the vendor cannot declaim conformance with the specification.

In the body of this Recommendation, the words shall, shall not, should, and may sometimes appear, in which case they are to be interpreted, respectively, as is required to, is prohibited from, is recommended, and can optionally. The appearance of such phrases or keywords in an appendix or in material explicitly marked as informative are to be interpreted as having no normative intent.

6 The overview of smart home service

Smart home is a home automation system in which a wide range of IoT devices in a home cooperate to provide intelligent controlling and monitoring functions for home users. Figure 1 illustrates the overall architecture of a smart home system. It consists of the following four major modules.

- **Smart home devices**: It consists of network enabled home appliances, sensors, actuators and switches, etc. They are basically aggregated by the smart home gateway as there are so many different wireless technologies and protocols adopted by different devices. These devices

connect to the physical world to collect data and send it to the smart home gateway, receive the device control command and execute it.

- **Cloud server**: Since the cloud has strong CPU power and unlimited storage capacity, large quantities of data such as user information, device information and service profile are usually stored in the cloud. Also, with a powerful CPU, advanced services such as linkage service, that interwork with other eco-systems are usually executed in the cloud. The decision will then be forwarded to a smart home gateway which in turn translates the command to the appropriate format and sends it to the corresponding device. Besides, user management and authentication, device management (DM), and other advanced services are usually implemented in the cloud.
- **User app**: Smart terminals such as smartphones and tablets usually have good process power and a friendly user interface. With them, users interact with the smart home system through client applications running over them, perform all kinds of control such as turning the device on/off, checking the status of the device, and setting timing service, etc.
- **Smart home gateway (SHGW)**: Smart home gateway plays a very important role in the system. It connects various smart home devices, provides hardware interfaces of various smart home communication protocols, runs communication protocols, performs protocol conversion and bridging, realizes device discovery and authentication access, and realizes the interaction between the user control terminal and the cloud server, such as requesting the device profile and service profile from the cloud server and sending the control command to the target device to realize the application service requested from the users.

First, every smart device in a home needs to be discovered and connected to the network that is formed by the gateway device, the process is called pairing or onboarding. During the process, the gateway and the device exchange some basic information such as manufacturer, device type identification and a list of the names of application profiles that are used in the device. If a new type of device is discovered, the gateway connects to the device profile server and downloads the device profile according to the device type. By parsing this information from the downloaded file and storing it in a database, the gateway is able to know all the details of the application profiles of the new device. Thus, without any software update, the gateway can provide correct representational state transfer (REST)ful APIs for a client application to communicate with the new device. The software architecture of the gateway is composed of multiple software layers (modules) as shown in Figure 2.

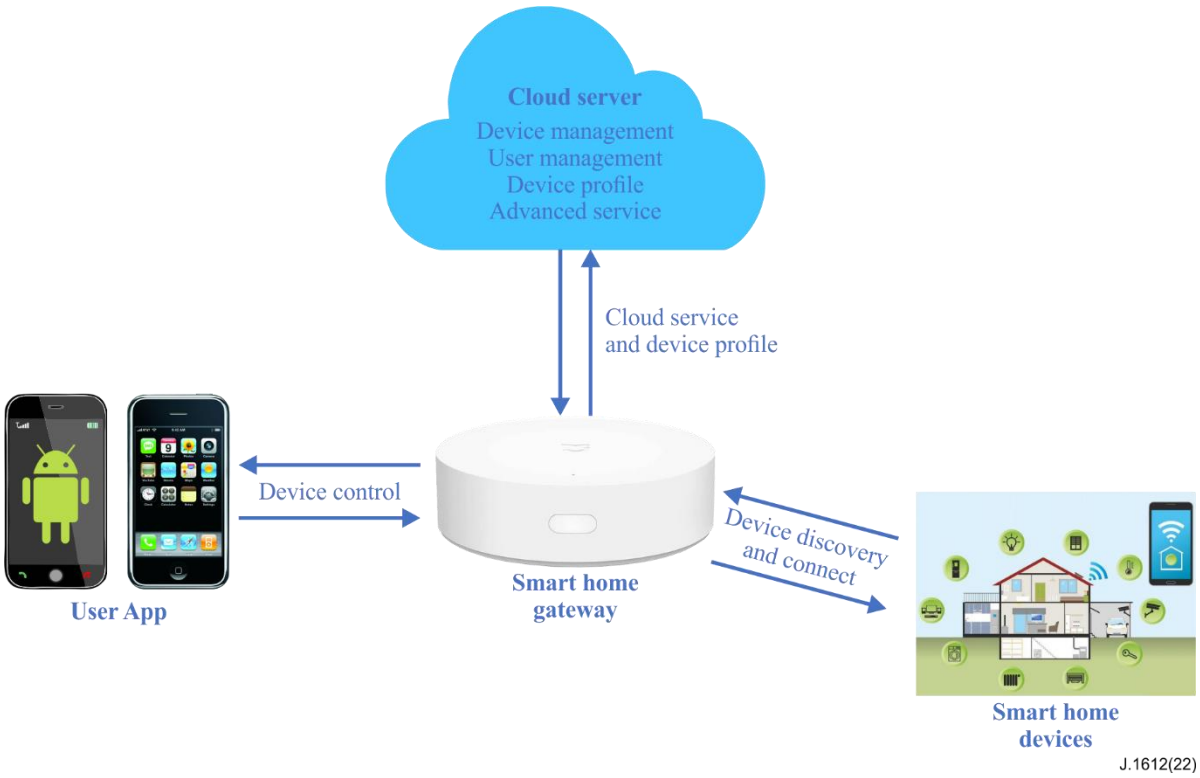

Figure 1 – The overall smart home system

7 Gateway software architecture

The software architecture of a smart home gateway is illustrated in Figure 2.

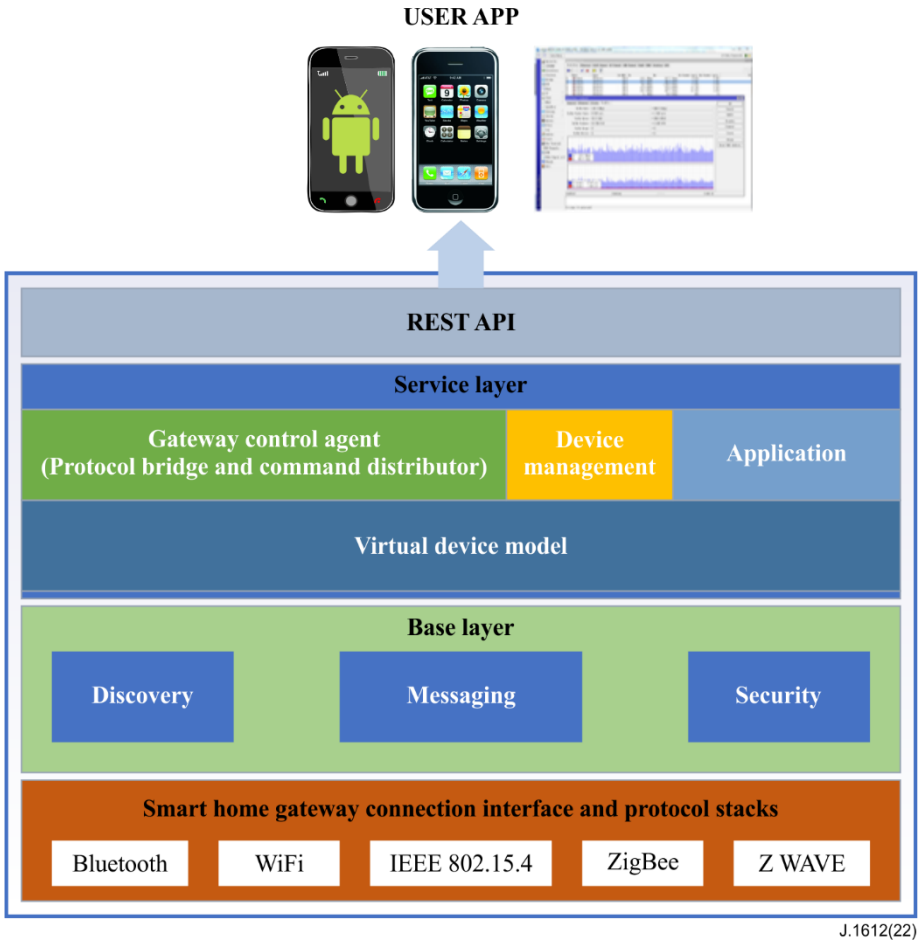

Figure 2 – Software architecture of a smart home gateway

The software architecture consists of multiple connection interfaces and corresponding protocol drivers (protocol stacks), base layer and service layer.

– The connection layer provides multiple wireless hardware interfaces and their corresponding driver (or protocols stack). With these interfaces, the various smart home devices can connect to the gateway.

– The base layer provides device discovery, messaging and security functions.

– Protocol bridge provides protocol translation between different protocols, such as Bluetooth low energy (BLE), Zigbee, Z-Wave, thread, etc.

– Service layer modules provide device management, protocol bridge and the control of the devices, onboarding service, etc.

7.1 A universal control platform

Designing a universal control platform for smart homes has been regarded as an important challenge. Based on the functional requirements of a smart home gateway, which are defined in [ITU-T J.1611], the general requirement for the platform is to provide a network communication technology that can access, or be compatible with a variety of smart device hardware, and also to adopt a virtualization technique which is responsible for shielding the communication details, as well as abstracting the system resources. From the application point of view, it is preferred to offer a software architecture to realize unified APIs for different device vendors and allow device drivers to be defined in, and composed by using a data description language such as JavaScript object notation (JSON) or extensible markup language (XML). By way of architecture design, it should be open and better to adapt a significant number of smart devices. Hence, the architecture needs to be able to handle

multiple protocols with unified control over heterogeneous networks, allowing developers to create applications to control diverse types of devices without any in-depth knowledge of the underlying protocols. For this purpose, a virtual device model is used for each physical device in the home to provide standard RESTful APIs for interacting with this virtual device. Besides resolving device compatibility, the proposed architecture can also solve the problem of the lack of a unified application interface in human-computer interaction systems due to the usage of heterogeneous home devices. It can also tackle the interoperability of multi-vendor devices to a certain extent. Due to the adoption of the cloud control environment, this architecture shows good adaptability in solving some important problems, including interconnection and collaborative control.

7.2 Communication interfaces and connectivity driver

The gateway usually consists of multiple different interfaces to provide the hardware compatibility to connect to various kinds of smart home access technologies. The typical wireless technologies used in Smart home services are Zigbee, Wi-Fi, Bluetooth, thread, Z-Wave, etc. The connectivity driver or protocols stack is the lowest layer of the system. It provides a software interface to ensure that the gateway software can interact with the hardware. The gateway hardware usually consists of a single-board computer, which includes a micro-processor, storage and communication modules e.g., Wi-Fi, Zigbee, Z-Wave, or Bluetooth low energy (BLE). Each type of communication module needs a particular driver to connect the hardware and the upper software layer. The driver is responsible for communicating with the physical devices.

7.3 Virtual device model

In order to provide unified control over heterogeneous networks, each physical device is abstracted into a logical device, which is called a virtual device. The main purpose of using the virtual device model is to enable developers to deal with any type of device without having a deep understanding of the underlying protocol or hardware design of the device.

In a virtual model, user applications can interact with these virtual devices through standard RESTful APIs. The description, status and function of each device are stored in the database. Since it is very complicated to store these different types of data in a traditional database, especially for unstructured data, not only SQL (NoSQL) is recommended to be used as the gateway database.

The virtual device model provides the following advantages:

i) It eases the creation of additional services or applications based on the virtual device approach. For example, by adding user permission information, security functions such as access control can be built.

ii) It simplifies the communication between devices. As the information of physical devices can update the device image in the gateway in a proactive way, virtual devices can accurately reflect the current state of physical devices. Therefore, the communication interaction between heterogeneous devices can be carried out through the virtual devices in the gateway system. This not only reduces the traffic and overhead between smart home devices but also increases the scalability of the gateway. For example, the actuator can obtain sensor data by simply querying the database without communicating with the sensor.

iii) It improves the data usage efficiency. The data information of virtual devices can be used to develop stateful applications, such as asynchronous event notification, publish / subscribe messages and cloud-based data analysis / storage services. Various sensor data from the device can be stored in the cloud server to be analysed and processed by the server, while users or devices can query these data remotely through the cloud server.

7.4 Device profile and profile server

A smart home system usually has the following characteristics:

i) Smart home devices usually have limited CPU capacity, memory and power resources. As a result, smart home devices can only use low-power communication protocols, such as Zigbee, Z-Wave, Bluetooth low-power (BLE) or thread.

ii) Except for thread, these protocols are not IP based. They use specific application layer design. In this case, all the functions of the device, such as open, close and sending notifications, are clearly defined in the protocol. Developers must follow the protocol specification to develop applications. Protocols usually organize a set of related functions into an application profile, which contains the details required to communicate with the device.

iii) In addition to standard configuration files, these protocols also support option fields for manufacturers to define their own options for special purposes or special applications. The advantage of this protocol design is that it makes developing applications more efficient and easier. On the other hand, it increases the processing load of the gateway. Specifically, the gateway needs to know all the application configuration files for each smart device in order to communicate normally with the smart devices.

iv) In order to coordinate applications and provide users with a seamless experience, the gateway needs to know additional information related to the device, such as the name and description of the product, as well as the pictures of the product. However, due to bandwidth constraints, they cannot be completely embedded in communication protocols for transmission and exchange. Therefore, the gateway manufacturer needs to pre-set the application configuration files and other information of various devices in the gateway. In other words, the gateway must know the information of all devices it will connect to be compatible with them. As new devices and applications emerge endlessly, the gateway software needs to continuously be upgraded to support new devices or applications, which becomes an impossible task for gateway manufacturers in reality.

To fit these characteristics, a profile needs to be defined for each device. The profile is stored in the profile server. The device's profile will be dynamically downloaded when it is put in use. In a smart home system with a cloud server, as shown in Figure 1, the application profile server is deployed in the cloud. The gateway equipment does not need to pre-set all the devices and their application profiles when the gateway is manufactured. After the smart home device manufacturer releases a new device or a new application, it only needs to upload the device and the corresponding application profile to the profile server. Then the existing gateway can support the new device and related applications by dynamically downloading the profile.

The workflow of the gateway supporting a new device is as follows:

i) A new device attempts to access the gateway and the gateway does not find its profile in the memory;

ii) The gateway sends a request to the profile server;

iii) The profile server queries the database to obtain the profile of the new device and sends it to the gateway;

iv) The gateway stores the profile in the gateway memory.

With this new profile, the gateway can support the new device and its application. In this architecture, JSON is chosen as the data format of the device profile. Other software modules of the gateway such as device database and RESTful API also use the JSON format.

This design reduces the cost of converting data between components. JSON has another advantage. It is a lightweight and has a readable format. The profile usually consists of two parts: general information and specific information. The general information includes the identification and description of the device, device type, protocol type, etc. The gateway can determine how to deal with the specific information in the device configuration file according to the basic information. For example, the information in the configuration file of Zigbee and Z-Wave devices can be interpreted differently. The specific information may include status and commands used to communicate with

the device, which are composed of status information, commands, descriptions of status and commands, etc.

7.5 RESTful API

RESTful web services are a set of standard APIs used to control communication between applications (clients) and gateways (servers). The client application can be a traditional desktop, mobile, or web application. Since client applications run on powerful equipment, it is suitable to implement RESTful API through hypertext transfer protocol (HTTP) protocol. Using standard APIs, developers can customize applications for specific customers. In order to provide unified control of heterogeneous networks without an in-depth understanding of real devices, various APIs can be defined according to the type and function of the devices. For each type of device, a set of common APIs are defined in advance. For example, for lighting devices, the ON and OFF commands will be defined in their APIs. In addition, based on the device profile, the gateway can provide customized APIs for each device. For example, a special light bulb may have an API to provide the temperature in the home. Compared with simple object access protocol (SOAP) based web services, RESTful web services have become more popular because of its simplicity and good performance while supporting various data formats. Since RESTful web services supports JSON format, it is very convenient to exchange data between modules in the gateway. In addition, the control application and gateway are separated, so that each component can be developed independently which reduces the development time of the gateway products. If the gateway manufacturer provides the open RESTful API, it is easy for third parties to develop control applications based on RESTful API due to its simplicity. In short, using RESTful web service helps the development of a smart home gateway.

7.6 Gateway control agent

The gateway control agent consists of a protocol translator and a command distributor. The protocols translator translates commands from a user application into the respective command in the driver for each protocol. From an application viewpoint, developers do not need to know the details of the underlying protocols because they only interact with virtual devices via RESTful APIs. All requests from the RESTful API or cloud are passed to the gateway control agent, that detects which device the user wants to control, which protocol this device uses, and how to translate the user's command to the corresponding protocol's command. For example, if a user wants to turn on a living room light, they press the ON button in an application. The application then sends an HTTP request to the gateway, and the information from the RESTful API passes to the gateway control agent, which gets the network address of this living room light from the database and understands that this light uses the Zigbee protocol. In the end, the gateway control agent requests to send the light-on command from the Zigbee driver to the physical living room light.

7.7 Device management

The raw data generated by smart home devices have heterogeneity such as data structure, functionality, etc. To use the sensor data and access the devices, a device data abstraction (virtual device model) module is required. It converts raw data to a common data format defined by the virtual device model. It provides interoperability between heterogeneous devices. In addition, before being ready for service, every smart device in a home needs to be discovered and connected to the network that is formed by the gateway device (this process is called onboarding). The procedures of discovery and connection are different according to each protocol (Zigbee, Z-Wave, Wi-Fi), and they may require some actions from the user (e.g., to push buttons or input code). Through this process the gateway and the device exchange some basic information such as manufacturer, device type identification and a list of the names of application profiles that are used in the device. The device type identification information can be different according to the underlying network protocols. Similarly, each protocol also provides a different mechanism to obtain the device type identification information. In fact, device type identification information is mandatory to distinguish a specific

device type from others. If a new type of device is discovered, the gateway connects to the device profile server and downloads the device profile according to each device type. A virtual device will be created in the database after the onboarding procedure completion. Then the gateway will be able to support this new device in a home.

7.8 Cloud interface

Cloud is required to provide an interface to the gateway and user terminal to achieve many important functions and services in a smart home system. The interface between cloud and the gateway or the device uses constrained application protocol (CoAP) [IETF RFC 7252] over TCP protocol. The cloud is a logical entity to which the smart home gateway (SHGW) communicates via a persistent TLS [IETF RFC 8446] connection. The gateway will interact with the cloud to achieve the following major functions:

Communication agent: A gateway usually located in a private home network, there is a persistent connection between the gateway and the cloud as a communication tunnel which provides a user terminal (usually a smartphone) to communicate with the gateway remotely.

Account management: There is a logical entity present in the cloud, the gateway needs to work with this entity to achieve user and device registration, access token validation and handles sign-in and token-refresh requests, etc.

Device profile management: There is a logical entity present in the cloud called a profile server to manage device profiles and other device information. It saves all kinds of device profiles that an SHGW will support, and when a new device is on the market, the manufacturer will upload its profile onto this server. The server will then respond to the request from the SHGW about the device profile which is indexed by device ID. In addition, if the device makers want to add new features to their device, they can just update the corresponding device profile on the server. In this case, the gateway will update the profile to reflect the changes.

Advanced service: SHGW will work with cloud to achieve some advanced services. Due to powerful computing power and abundant storage space, some advanced services could also be implemented on cloud, such as timing service, linkage service and interworking with other smart home ecosystems, etc.

Bibliography

[b-ITU-T Y.4000] Recommendation ITU-T Y.4000/Y.2060 (2012), *Overview of the Internet of things*.

[b-ITU-T Y.4101] Recommendation ITU-T Y.4101/Y.2067 (2017), *Common requirements and capabilities of a gateway for Internet of things applications*.

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	Tariff and accounting principles and international telecommunication/ICT economic and policy issues
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	**Cable networks and transmission of television, sound programme and other multimedia signals**
Series K	Protection against interference
Series L	Environment and ICTs, climate change, e-waste, energy efficiency; construction, installation and protection of cables and other elements of outside plant
Series M	Telecommunication management, including TMN and network maintenance
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling, and associated measurements and tests
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks, open system communications and security
Series Y	Global information infrastructure, Internet protocol aspects, next-generation networks, Internet of Things and smart cities
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland
Geneva, 2022