Original scientific paper

Prototype Wireless Network for Internet of Things based on DECT Standard

Ivan V. Sinyavskiy, Igor M. Sorokin, and Andrei M. Sukhov, *Member, IEEE*

***Abstract* — This paper presents a software prototype of a wireless network for the Internet of Things (IoT) based on the DECT (Digital Enhanced Cordless Telecommunication) standard. It proposes an architecture for encapsulating commands from the most common IoT protocol, MQTT (Message Queuing Telemetry Transport), into SIP (Session Initiation Protocol) packets. A module is created to embed MQTT-SN (MQTT for Sensor Networks) packets into SIP packets. The module is developed in Go language using the built-in "net" library. Delivery of MQTT-SN packets to IoT devices is carried out using the SIP protocol. Source codes and instructions for installing the gateway can be found at https://github.com/iSinyavsky/mqtt-sn-sip-gateway.**

***Keywords* — DECT for IoT, encapsulation of IoT command, MQTT over SIP, SIP routing.**

I. Introduction

The Internet of Things has evolved in recent years from an exotic technology for connecting household appliances into an emerging industry that addresses numerous consumer, commercial, industrial and infrastructure challenges. However, the rapid development of such technology has led to a spectrum of problems. Within this spectrum, problems with the security of the technology and the network connectivity of IoT devices stand out.

Security flaws in the technology have led to the emergence of botnets of enormous size [1]. On the communications side, providers have also proved unprepared for the IoT boom [2]. Conventional communication systems for connecting ordinary users are ill-suited to IoT networks, as they were designed for very different purposes.

Network connectivity for IoT devices must have many features, including:

- Low power consumption.
- Longer network range, up to several hundred metres, which is considerably larger than that of WiFi or Bluetooth.
- High network bandwidth and high data transfer rates are not required.
- The range of wireless frequencies must be free and must not be subject to government regulation.

Currently, among wireless connections, WiFi leads the way, with cellular networks taking a fairly large share [3]. These technologies, however, do not meet all requirements and are quite costly to establish and operate.

For this reason, there are currently numerous attempts to develop new wireless communication standards that are fully adapted for connecting IoT devices. Specialized communication technologies generally use narrowband modulation. However, there are at least 7 such systems on the Russian market alone, but these are fundamentally different technologies for building IoT networks. There is no unified standard for wireless connection of IoT devices yet. In addition, all existing technologies are not fully open protocols.

LoRa (Long Range) technology [4] uses a proprietary modulation algorithm, so the chip for LoRa operation is proprietary and protected by a French patent. This is why this technology cannot be considered open source. The Sigfox platform [5] is also proprietary and provided as an off-the-shelf solution. The Russian technology Strizh [6] is based on the proprietary, narrow-band XNB (Extended Narrow Band) protocol and is fully controlled by the developer.

We have previously put forward the idea [7] of the possibility of reviving the old DECT technology and using it for IoT needs. This technology is currently experiencing a second birth due to a whole family of new standards adopted by ETSI (European Telecommunications Standards Institute) in 2020. [8]. Our report at TELFOR 2021 [9] lists the main advantages of DECT technology over existing and ongoing developments. These include the increased communication range, the unfilled frequency range, the energy-saving technology and the sufficient transmission speed.

The focus was on DECT-based communication circuits. Two possible ways of communication were pointed out. The first involves full TCP/IP connectivity at the IoT executive device using DECT. The second method assumes that the DECT handset supports VoIP (Voice over IP) technology. In this case, IoT control commands are encapsulated in VoIP packets on the IoT service server, transmitted to the DECT terminal, where they are retrieved. However, the idea of encapsulation was only suggested in the report and was not specified in any way.

Paper received May 14, 2022; revised July 11, 2022; accepted July 13, 2022. Date of publication August 05, 2022. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Grozdan Petrović.

This paper is revised and expanded version of the paper presented at the 29th Telecommunications Forum TELFOR 2021 [9].

The publication was prepared within the framework of the Academic Fund Program at the National Research University Higher School of Economics (HSE) in 2021 (Grant no.: 21-04-030).

Ivan Sinyavskiy is with HSE University, Tallinskaya, 34, Moscow, 34, 123458 (e-mail: ivsinyavskiy@edu.hse.ru).

Igor Sorokin is with HSE University, Tallinskaya, 34, Moscow, 34, 123458 (e-mail: imsorokin@hse.ru).

Corresponding author: Andrei Sukhov is with HSE University, Tallinskaya, 34, Moscow, 34, 123458 (e-mail: amsuhkov@hse.ru).

This paper focuses on how to solve the problem of encapsulating MQTT packets into VoIP packets and arranging the exchange of IoT control packets from the cloud server to the actuator. In this case, a wireless network of the DECT standard is used as the last mile.

The article not only analyses the basic technology of packet encapsulation, but also proposes an original solution, which is tested in experiments. Thus, we can talk about a software prototype of a wireless network for the Internet of Things based on the DECT standard. Unfortunately, our project is not yet supported financially, so we have not yet been able to present a complete software-hardware prototype, but the available groundwork is enough to implement it as soon as possible.

The article is implemented as follows. The second section will outline the basic approaches for implementing wireless communication for IoT devices based on DECT technology. The third section describes the software implementation of encapsulation, the fourth section describes routing using the SIP protocol. Conclusions are then drawn and our next steps for the practical implementation of the developed prototype are described.

II. Basic Provisions for using DECT as a Wireless Network for IoT Devices

When creating a new technology, there are several important points to consider. Only the successful implementation of these provisions will ensure that the new technology will not be stillborn, will be widely used, and that the money spent on its development will pay off.

The main critical provisions for new development are

- The resources spent, including financial resources
- Time needed for development
- Simplicity of the proposed solution and its compliance with the technical requirements

These are the points that have guided our development.

To keep the design as simple as possible and shorten the development time, no changes were made to the DECT technology. It is accepted as the basis, and its subscriber devices must be based on minicomputers with a Linux or Android operating system. This is the only requirement for the hardware platform.

Our innovations can be summarized in two points:

1. encapsulating IoT control packets (MQTT) in VoIP in the control cloud and retrieving them back on the user device

2. SIP protocol is used for routing. That is, an extra layer is added to the protocol stack. However, we have to allocate two sections of the route, on each of these sections the encapsulation scheme will be different. The first section from the control cloud to the DECT base station is called the backbone section. In this section all network devices transfer data according to the TCP/IP protocol stack, each device has an IP address. The protocol encapsulation diagram is shown in Fig. 1. The corresponding communication scheme is discussed in more detail in section 4. In our solution, we used a specific SIP software implementation in the form of the freely distributed Asterisk solution. Note that all elements involved in the complete communication scheme from Section 4 are assigned specific SIP addresses. It is on the basis of these addresses that routing takes place. This principle leads to the fact that the shortage of IPv4 addresses can be done away with, as they are not needed for the actuators. But there is another positive effect of this approach.

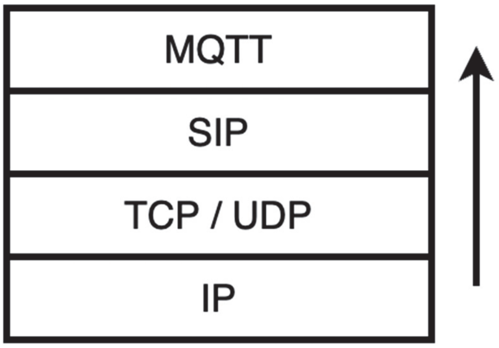


Fig. 1. Schematic of the proposed protocol stack on the backbone.

Since the IoT executive device is connected via DECT, and the full TCP/IP stack is not up, it will be very difficult to hack. It is also difficult to use this device to conduct network attacks.

The proposed solutions will be described in more detail in the following sections, here we would like to focus on the financial and labour costs involved in this development. This project is at this stage a personal initiative of the participants and is not yet supported by research funds. Less than a year has passed from idea to implementation. The participants are one professor, one postgraduate student and 4 students, who are carrying out educational projects on the same topic.

When comparing the costs and timelines of prototyping, the other technologies listed in the introduction required large teams. And the term of this work was not less than two years, and the costs amounted to at least hundreds of millions of dollars for foreign companies, while Russian companies spent tens of millions of rubles.

This difference can be explained by the fact that we were able to organise the research for this project as exploratory research. This is research that involves not only the search for new knowledge, as in the case of fundamental research, but also a specific area of application of the result found. It is precisely this type of research that should be a priority in Russia.

III. Software Implementation of Encapsulation

F The first of the two key technologies for building a prototype wireless network for the Internet of Things based on the DECT standard is packet encapsulation of IoT control protocol packets into VoIP packets.

IoT protocols are plentiful, we have chosen one of the most commonly used protocols MQTT (Message Queue Telemetry Transport) [10] to investigate encapsulation methods. It is an open data exchange protocol designed to transmit packets over a WAN when there are bandwidth constraints. The basic principle of the protocol is a subscription scheme with asynchronous operation, which is shown in Fig. 2.

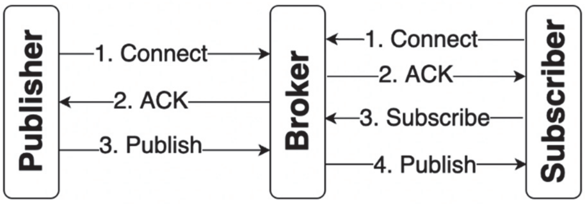


Fig. 2. Diagram of the MQTT operation.

Each MQTT command is transmitted in a separate message. In order to implement our idea, it is necessary to encapsulate these messages in SIP packets. SIP is a session establishment protocol used in VoIP. Encapsulation scheme on the trunk channel is shown in Fig. 1. The encapsulation scheme on the DECT radio channel (last mile) is quite different. The last mile is the section from the base station to the DECT handset combined with the IoT executive device. In this section, a DECT frame is transmitted over the radio channel. This is where the SIP packet with the MQTT command is encapsulated.

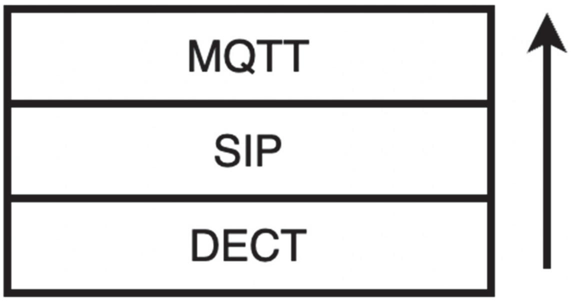
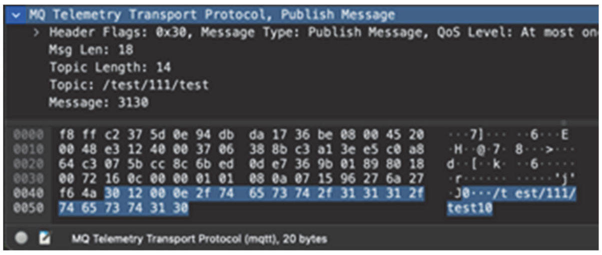


Fig. 3. Encapsulation scheme at the last mile.

The payload length in MQTT can be up to 456 bytes in one packet and 2 bytes per fixed header. The SIP protocol can send up to 1300 bytes without fragmentation. It is worth considering the specifics of IoT messages transmitted via the MQTT protocol. For example, the size of an MQTT packet published on topic /test/111/test with value 10 is 20 bytes (Fig. 4).



```
⌄ MQ Telemetry Transport Protocol, Publish Message
  > Header Flags: 0x30, Message Type: Publish Message, QoS Level: At most on
    Msg Len: 18
    Topic Length: 14
    Topic: /test/111/test
    Message: 3130

0000  f8 ff c2 37 5d 0e 94 db  da 17 36 be 08 00 45 20   ···7]··· ··6···E
0010  00 48 e3 12 40 00 37 06  38 8b c3 a1 3e e5 c0 a8   ·H··@·7· 8···>···
0020  64 c3 07 5b cc 8c 6b ed  0d e7 36 9b 01 89 80 18   d··[··k· ··6·····
0030  00 72 16 0c 00 00 01 01  08 0a 07 15 96 27 6a 27   ·r······ ·····'j'
0040  f6 4a 30 12 00 0e 2f 74  65 73 74 2f 31 31 31 2f   ·J0···/t est/111/
0050  74 65 73 74 31 30                                  test10

MQ Telemetry Transport Protocol (mqtt), 20 bytes
```

Fig. 4. MQTT packet size in WireShark software.

The following message types are embedded in the MQTT protocol:

- connect (access/connection establishment);
- disconnect (disconnect)
- publish: (publication of information in a topic)
- subscribe: (subscription to the topic)
- unsubscribe: (publication of information on the topic)
- unsubscribe: unsubscribe from topic.

In SIP there is a "Message" request, which allows the transfer of SMS messages between subscribers, the request consists of a SIP header and attached data, usually in text/plain format. The MQTT data can be transferred within the SIP Message, having previously prepared it for the required format, for example, by converting the payload bytes into hexadecimal string representation.

The next step to solve the encapsulation issue is to create a module to allow packing MQTT packets into SIP packets. Two alternative encapsulation options were tested.

According to the first variant, development was proposed to be done in Python, using the Scapy library. This tool allows to create your own network packets or capture and modify them. However, this way turned out to be a dead end.

The second way led to success. The module was developed in Go language using the built-in "net" library. This module is a gateway that receives packets of one type, performs processing and sends packets of another type: MQTT -> SIP, SIP -> MQTT. Both gateways are located between the broker and the subscriber with Fig. 2.

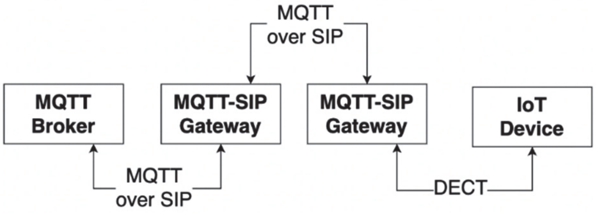


Fig. 5. Encapsulation diagram.

In this case, the MQTT-SIP Gateway acts as a proxy server handling request. The same gateway has to be implemented on the subscriber terminal. The final communication scheme is shown in Fig. 5.

Interim testing during development revealed a problem that prevented the encapsulation module from working correctly. SIP packets are mostly sent over the transport protocol UDP, while MQTT is based on TCP and works on a subscription basis. Therefore, an alternative solution has been developed to support a new version of the MQTT protocol which is abbreviated as MQTT-SN.

MQTT-SN is a MQTT based protocol developed especially for sensor networks [11]. Benefits include reduced packet size and operation over the UDP transport protocol. The reduction in packet size is achieved by reducing service information. In order to work with MQTT-SN protocol, standard MQTT broker is used on IoT services server, with addition of additional element MQTT-SN Gateway. Its task is to transform the MQTT message into MQTT-SN and vice versa. This protocol is supported by Eclipse, which develops the most used version of the standard MQTT broker.

Thus, this section of the paper describes the creation of a software module that encapsulates MQTT-SN packets into SIP packets, sends them onwards to the destination and retrieves them at the end of the route, according to the scheme shown in Fig. 5. Note that the resulting communication scheme is two-way, packets can be transmitted both from the publisher to the subscriber through the broker and in the opposite direction.

IV. Routing Scheme

Our proposed communication scheme has an essential feature related to encapsulation, the use of SIP packets and the DECT wireless section as the last mile. Therefore, it is necessary to use a SIP switch to deliver packets between the MQTT-SIP gateways. For the delivery of SIP packets, the open-source communication platform Asterisk was chosen, which implements all the features of modern digital telephony. This choice was due to its versatility, most of the VoIP equipment is supported by this platform.

Fig. 6 shows the scheme of SIP telephony software blocks, which performs encapsulation and delivery of SIP packets.

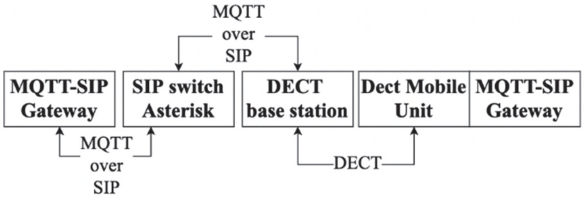


Fig. 6. Schematic diagram of the software blocks.

A SIP switch, a DECT base station and a subscriber unit are placed between the two MQTT-SIP gateways. These components ensure the delivery of the SIP packet between the gateways, with all Software Blocks in Fig. 6 receiving SIP addresses.

An interesting question is the hardware implementation of Fig. 6 scheme. The number of hardware components could be reduced to three. In our opinion, it is possible to combine MQTT-SIP gateway and SIP switch, as well as subscriber device, MQTT-SIP gateway and IoT device on a single platform. This combination has an advantage over having a SIP switch and a DECT base station on the same platform, because in the first case the switch will be able to serve several base stations.

Again, it must be emphasized that the complete TCP/IP protocol stack is raised at the point from the broker to the DECT base station. Whereas on the last mile only SIP packets can be transmitted as a DECT wireless segment. This means that most TCP/IP application protocols such as telnet, ssh, http etc. will not be available. This fact will make it very difficult to use IoT devices as elements of botnets, as both managing and attacking them over the network will be very difficult. New network attack mechanisms need to be invented.

To make this technology cheaper, a single chip, including a DECT module and an IoT executive module (Arduino, for example) with hardware support for the MQTT-SIP gateway, must be created. In principle, the migration of this technology to a hardware platform needs to be worked out.

V. Conclusion

The paper presents an implementation of last mile wireless technology for IoT. It has been proposed to use the old DECT wireless technology as a similar technology. This technology is currently experiencing a second birth due to a whole family of new standards adopted by ETSI in 2020.

This article lists the main advantages of DECT compared to existing and ongoing developments. These include the increased communication range, the unfilled frequency range, the energy-saving technology, and the sufficient transmission speed.

The study also explored the possibility of encapsulating IoT commands into VoIP (SIP) packets. The architecture of encapsulation is proposed, in particular, the theoretical possibility of encapsulating commands of the most common in the IoT protocol MQTT in SIP packets is justified.

As a result, a module has been created that allows MQTT-SN packets to be embedded in SIP packets. The module is developed in Go language using the built-in "net" library. The module is a gateway that receives packets of one type, processes them and sends packets of another type: MQTT -> SIP, SIP -> MQTT. Thus, it is possible to talk about a software prototype of a wireless network for the Internet of things based on the DECT standard. Source codes and instructions for installing the software for the gateway can be found at the link [12].

The success of the prototype is due to innovations, which can be summarized in two points. The first involves encapsulating IoT control packets (MQTT) in VoIP in the control cloud and retrieving them back at the user device. The second point of novelty is the use of the SIP protocol for routing. The proposed solution uses a specific software implementation of SIP telephony, the freely distributed Asterisk solution.

References

[1] C. Kolias, G. Kambourakis, A. Stavrou and J. Voas, “DDoS in the IoT: Mirai and Other Botnets,” in *Computer*, vol. 50, no. 7, pp. 80-84, 2017.

[2] S. Al-Sarawi, M. Anbar, K. Alieyan and M. Alzubaidi, “Internet of Things (IoT) communication protocols: Review,” *2017 8th International Conference on Information Technology (ICIT)*, 2017, pp. 685-690..

[3] S. Li, L. Da Xu and S. Zhao, “5G internet of things: A survey,” *J. Ind. Inf. Integration*, vol. 10, pp. 1-9, 2018.

[4] A. Augustin, J. Yi, T. Clausen, and W. M. Townsley, “A study of LoRa: Long range & low power networks for the internet of things,” *Sensors* 16, no. 9 (2016).

[5] A. Augustin, J. Yi, T. Clausen, and W. Townsley, “A Study of LoRa: Long Range &amp; Low Power Networks for the Internet of Things,” *Sensors*, vol. 16, no. 9, p. 1466, Sep. 2016.

[6] A. Lavric, A. I. Petrariu and V. Popa, “SigFox Communication Protocol: The New Era of IoT?,” *2019 International Conference on Sensing and Instrumentation in IoT Era (ISSI)*, 2019, pp. 1-4.

[7] S. Petrenko, A. Petrenko, K. A. Makoveichuk, and A. Olifirov, “Development of a Cyber-Resistant Platform for the Internet of Things Based on Dynamic Control Technology,” In *International Conference on Futuristic Trends in Networks and Computing Technologies*, pp. 144-154. Springer, Singapore, 2020.

[8] A. Sukhov, I. Sorokin, and D. Meil, “New life for cordless communication, old regrets for software projects,” *Communications of the ACM 64*, no. 10 (2021).

[9] DECT, ETSI. "New Radio (NR); Part 1: Overview; Release 1." European Telecommunications Standards Institute, Technical Specification (TS) 103 (2020).

[10] I. M. Sorokin, L. A. Romanov, and A. M. Sukhov, “DECT standard as wireless IoT technology,” In *2021 29th Telecommunications Forum (TELFOR)*, pp. 1-4. IEEE, 2021.

[11] Standard, O.A.S.I.S. "MQTT version 3.1.1." URL http://docs. oasis-open. org/mqtt/mqtt/v3 (accessed: 15.03.2022).

[12] A. Stanford-Clark and Hong Linh Truong, “Mqtt for sensor networks (mqtt-sn) protocol specification,” *International business machines (IBM) Corporation version 1*, no. 2 (2013): 1-28.

[13] https://github.com/iSinyavsky/mqtt-sn-sip-gateway