УДК 004.72
DOI: 10.25559/SITITO.18.202203.625-633

Original article

Development of Cloud Platform for Controlling Internet of Things Devices

D. A. Perepelkin, **D. D. Tkachev***
Ryazan State Radio Engineering University named after V.F. Utkin, Ryazan, Russian Federation
Address: 59/1 Gagarina St., Ryazan 390005, Russian Federation
* fenix784@mail.ru

Abstract
Currently, solutions based on the Internet of Things are becoming widely in demand. The concept of the Internet of Things implies the construction of a software-defined network (SDN) of physical devices with integrated mechanisms of interaction both among themselves and with a cloud platform, a software system and objects of the outside world. The aim of the work is to develop a cloud platform and a visual software system for configuring Internet of Things devices. The paper proposes a four-level architecture of a software-defined network, including an additional layer of Internet of Things devices containing actuators for executing commands and sensors for collecting environmental parameters. To aggregate data about the network structure, as well as to obtain information from the end devices of the Internet of Things and store their configuration, a cloud platform has been developed that allows using an external REST API interface and a deployed socket server to interact with the visual software system and end devices of the Internet of Things. A visual IoT Map software system has been developed for configuring a software-defined network of Internet of Things devices, visualizing the network topology and data received from end devices. Special attention is paid to the mechanisms of interaction between the visual software system, the cloud platform and the end devices of the Internet of Things.

Keywords: Internet of Things, software-defined networks, cloud platform, network device scanner, visual software system, network architecture

Funding: The work was carried out with the financial support of the grant of the President of the Russian Federation no. MD-3201.2022.1.6 "Design and Development of Methods, Algorithms and Technology of Software-Configurable Cloud Platforms of the Internet of Things".

Conflict of interests: The authors declare no conflict of interest.

For citation: Perepelkin D.A., Tkachev D.D. Development of a Cloud Platform for Controlling Internet of Things Devices. *Modern Information Technologies and IT-Education.* 2022;18(3):625-633. doi: https://doi.org/10.25559/SITITO.18.202203.625-633

Оригинальная статья

Разработка облачной платформы управления устройствами Интернета вещей

Д. А. Перепелкин, Д. Д. Ткачев*
ФГБОУ ВО «Рязанский государственный радиотехнический университет им. В.Ф. Уткина», г. Рязань, Российская Федерация
Адрес: 390005, Российская Федерация, г. Рязань, ул. Гагарина, д. 59/1
* fenix784@mail.ru

Аннотация
В настоящее время широко востребованными становятся решения на базе Интернета вещей. Концепция Интернета вещей подразумевает построение программно-конфигурируемой сети (ПКС) физических устройств с интегрированными в них механизмами взаимодействия как между собой, так и с облачной платформой, программной системой и объектами внешнего мира. Целью работы является разработка облачной платформы и визуальной программной системы конфигурирования устройств Интернета вещей. В работе предложена четырехуровневая архитектура программно-конфигурируемой сети, включающая дополнительный уровень устройств Интернета вещей, содержащий исполнительные устройства для выполнения команд и датчики для сбора параметров окружающей среды. Для агрегирования данных о структуре сети, а также получения информации с конечных устройств Интернета вещей и хранения их конфигурации разработана облачная платформа, позволяющая с помощью внешнего интерфейса REST API и развернутого сокет-сервера осуществлять взаимодействие с визуальной программной системой и конечными устройствами Интернета вещей. Для конфигурирования ПКС устройств Интернета вещей, визуализации топологии сети и данных, получаемых с конечных устройств, разработана визуальная программная система IoT Map. Особое внимание в работе уделено механизмам взаимодействия между визуальной программной системой, облачной платформой и конечными устройствами Интернета вещей.

Ключевые слова: Интернет вещей, программно-конфигурируемые сети, облачная платформа, сканер устройств сети, визуальная программная система, сетевая архитектура

Финансирование: Работа выполнена при финансовой поддержке гранта Президента Российской Федерации для государственной поддержки молодых российских ученых – докторов наук № МД-3201.2022.1.6 «Разработка и развитие методов, алгоритмов и технологии построения программно-конфигурируемых облачных платформ Интернета вещей».

Конфликт интересов: авторы заявляют об отсутствии конфликта интересов.

Для цитирования: Перепелкин Д. А., Ткачев Д. Д. Разработка облачной платформы управления устройствами Интернета вещей // Современные информационные технологии и ИТ-образование. 2022. Т. 18, № 3. С. 625-633. doi: https://doi.org/10.25559/SITITO.18.202203.625-633

Introduction

The Internet of Things (IoT) is a concept of building a system of physical devices interconnected with each other and the external environment, which allows collecting, transmitting and processing data practically without the need for human participation. Currently, the concept of the Internet of Things has a fairly wide range of applications, automating processes from various fields of human activity. Internet of Things software systems allow you to monitor, visualize and manage data from connected devices.

Software-defined networks (SDN) are used to deploy Internet of Things systems. In the simplest case, the architecture of a software-defined network can be represented as a set of software systems and hardware switches connected to a common OpenFlow controller. This architecture is based on three levels: the network application level, the management level, and the infrastructure level.

The infrastructure level contains switches, transmission channels and information processing systems and performs data aggregation and transformation functions. The management level is software tools that implement various mechanisms for managing infrastructure-level devices. The network application level includes a set of applications that allow you to flexibly and efficiently visualize, analyze data and manage network infrastructure.

In recent years, research and development using the concept of the Internet of Things and the technology of software-configurable networks have been gaining popularity massively. In this paper, a four-level architecture of IoT devices is proposed, a wireless network device scanner and a cloud platform for collecting, processing data and managing IoT devices in real time are developed, and a visual software system for configuring IoT devices is described.

Theoretical information

The paper[1] provides detailed information about the ecosystem of the Internet of Things, considers the architecture and technologies for building a software-defined network of Internet of Things devices. Work[2] contains detailed information about the basic principles of the construction, design and support of the SDN. Various approaches to the technical implementation of the Internet of Things concept, protocols and data transfer technologies are discussed in[3] [1-5]. The creation of new devices within the framework of the Internet of Things concept, their connection using Wi-Fi technology and data exchange with a cloud platform are discussed in[4]. The basics of cloud computing, existing models for building and deploying cloud platforms, virtualization technologies and information security of cloud infrastructure are described in[5]. Optimization of data flows, network configuration using software applications are described in[6] [6-7]. The paper [8] considers the conceptual approach of dynamic traffic generation of software-defined telecommunication networks with load balancing. The visual environment and software infrastructure for distributed data processing are presented in [9]. Approaches to dynamic traffic management in the cloud infrastructure of software- defined networks and data centers are considered in [10-12]. In [13], ETSI's proposals in the field of the Internet of Things and the principle of operation of the oneM2M standard are considered. Approaches to international standardization and programming of the Internet of Things, issues of cybersecurity of the Internet of Things are considered in [14]. In [15], various programming models for Internet of Things applications are proposed, depending on their complexity, scale and types of supported devices. Various application-level interfaces for software-defined networks are considered in [16]. In [17], various cloud platforms for the implementation of software systems based on the concept of the Internet of Things are considered, their main capabilities, advantages and disadvantages are identified. Application models, the role of the Internet of Things platform, application architecture based on the Internet of Things concept are discussed in [18]. The paper [19] describes a smart home management system with an integrated cloud environment that provides remote device management and data monitoring capabilities [20-22]. The paper [23] discusses the use of cloud platforms in building systems based on the concept of the Internet of Things [24]. The analysis of the above works has shown the relevance and necessity of solving the scientific problem of building a software-defined network of Internet of Things devices with support for cloud computing and the ability to configure devices and network services [25].

Architecture of SDN of Internet of Things devices

Since the basic elements in the concept of the Internet of Things are finite physical devices, in order to combine SDN technology and the concept of the Internet of Things in the development of software systems in the classical architecture of a software-configurable network, it is proposed to include another level – the level of Internet of Things devices, thereby obtaining a four-level SDN architecture.

The management level is represented by a cloud platform, which, in addition to performing data processing and storage functions, network management and end devices, implements software integration methods with a visual software system.

The level of Internet of Things devices includes sensors that collect, process and transmit information about the surrounding world over the network, and actuators that collect information, perform mechanical work and control processes.

[1] Lea P. Internet of Things for Architects. Packt Publishing; 2018. 524 p.

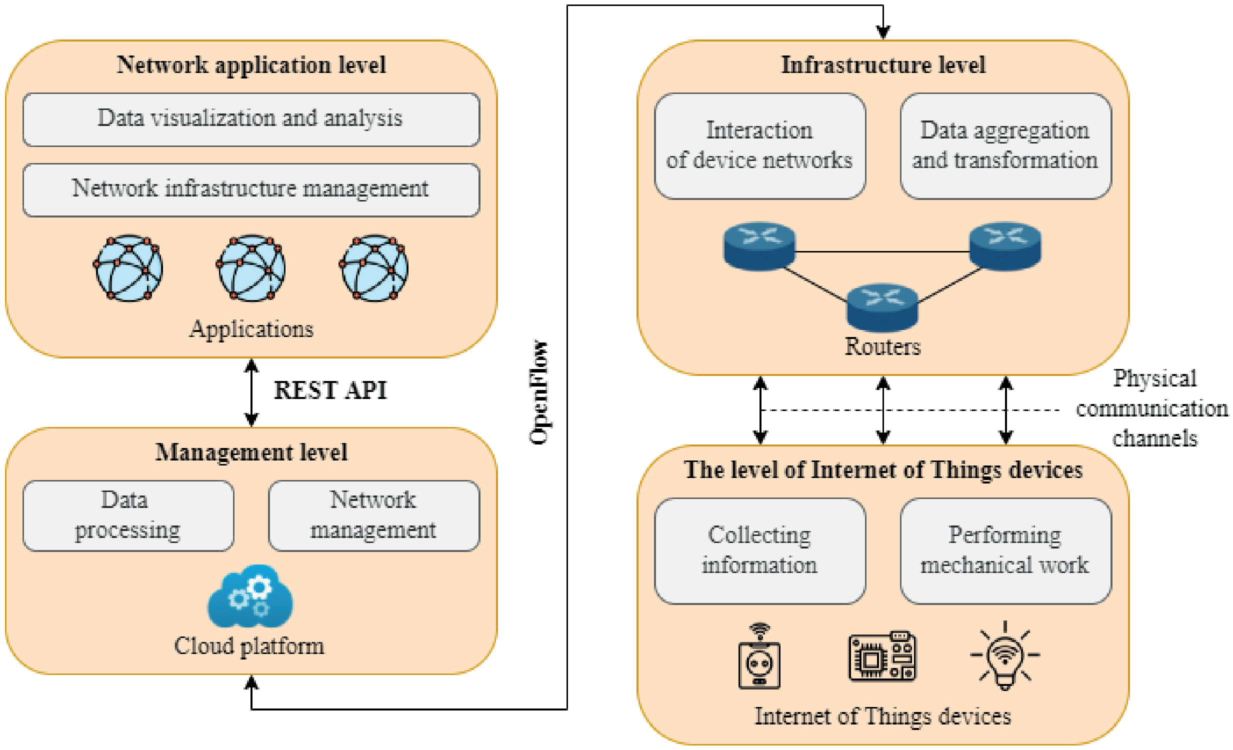
[2] Koryachko V.P., Perepelkin D.A. [Software-Defined Networks: University Textbook]. Moscow: Hotline-Telecom Publ.; 2020. 288 p. (In Russ.)

[3] Roslyakov A.V., Vanyashin S.V., Grebeshkov A.Yu. [Internet of Things: University Textbook]. Samara: PSUTI; 2015. 135 p. (In Russ.)

[4] Petin V.A. [New features of Arduino, ESP, Raspberry Pi in IoT projects]. St. Petersburg: BHV-Petersburg; 2022. 320 p. (In Russ.)

[5] Andreevsky I.L. [Cloud computing technologies: University Textbook]. St. Petersburg: SPbSUE Publ.; 2018. 79 p. Available at: https://www.elibrary.ru/item.asp?id=27543361 (accessed 13.06.2022). (In Russ.)

[6] E˘gilmez H.E. Adaptive Video Streaming over OpenFlow Networks with Quality of Service : Thesis for Degree of Master Science in Electrical and Electronics Engineering. Koc University; 2012. 80 p. Available at: http://www-scf.usc.edu/~hegilmez/hilmi_ms_thesis.pdf (accessed 13.06.2022).



F i g. 1. Four-layer architecture of a SDN of Internet of Things devices

Development of a wireless network device scanner

The wireless network device scanner (network scanner) scans the network to which the device is connected and detects all clients connected to this network. Identification of each client in the network occurs using its IP and MAC address. Ping and arp utilities based on ICMP and ARP protocols are used to identify active devices. The ping utility sends an ICMP internetwork control message protocol echo request packet to the specified client and waits for an ICMP echo response. The arp utility allows you to determine the client's MAC address by a known IP address.

The *get_router_ip()* and *get_server_ip()* methods of the network scanner are used, respectively, to obtain the local IP addresses of the router and the device on which the scanner is running. The *get_global_ip()* method returns the external (global) IP address of the entire node. Based on the router's local IP address, the subnet identifier is determined and a pool of available IP addresses of devices in the network is formed.

To execute the ping utility, the *ping_device()* method is used, which takes the target IP address as an input parameter and determines the availability of the device at this IP address. This method also allows you to determine the data transfer delay to the device and its type. The *connect_pyp100()* method determines whether the device being scanned is a TP-Link Tapo P100 smart plug, and in case of a successful connection to it, it rotates the current state of the plug. The *get_devices()* method implements the execution of the arp utility and returns a dictionary of correspondences of IP and MAC addresses of devices on the network. Multithreading is implemented in the network scanner to speed up device scanning.

Since the format of ping and arp utilities is different in Windows and Linux operating systems, the network scanner determines the type of operating system at startup and selects the required format for further execution of utilities.

In addition, the scanner implements sending information about the devices of this network to the cloud platform using a socket – a bidirectional connection over which data is transmitted. The *send_messages()* method sends data to the cloud via a socket, and the *receive_messages()* method is responsible for listening to incoming messages. If the cloud is unavailable, the scanner cyclically attempts to establish a connection with it in the *do_connection_attempt()* method until the connection is established.

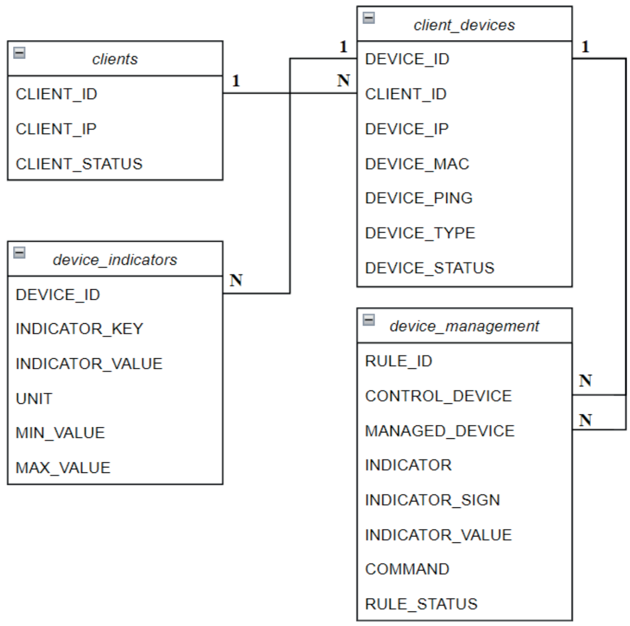
All information about the network is sent to the cloud in the form of a json file containing information about IP addresses, MAC addresses, delays and device types of this network.

To control the TP-Link Tapo P100 smart plug, the network scanner contains the *turn_on()* and *turn_off()* methods, which are responsible for turning the plug on and off, respectively. The IP address of the managed outlet and the type of command are contained in a json object received by the network scanner from the cloud platform. If the target plug is unavailable at a particular time, the scanner will continue to cycle attempts to connect to it until the connection is established and the command is executed.

Development of the Internet of Things cloud platform

The cloud platform of the Internet of Things aggregates data about the overall network structure, stores information about the end devices of the Internet of Things, generates commands depending on the network configuration and transmits them to the executive device, implements external APIs for integration with the visual software system.

A database has been developed to store information about all nodes and end devices of a software-configured network of Internet of Things devices in a cloud platform. In the database, the identification of nodes of a software-defined network is carried out using their global IP address. IoT end devices are identified using a MAC address.



F i g. 2. Schema database of the Internet of Things cloud platform

To interact with the database in the cloud platform, the *select()* and *insert_or_update()* methods are implemented. Each of the methods takes as input values a parameterized SQL string and a tuple of parameters required to execute the query. The *select()* method is used to extract data from the database and returns the tuple of values received after executing the query. The *insert_or_update()* method is used to perform the operation of inserting, updating and deleting records in the database. A connection pool is used to connect to the database, which caches connections in memory for reuse without having to create a new connection to the database from scratch. This approach provides a significant increase in performance when working with the database.

A socket server is implemented in the cloud to receive data from devices collecting Internet of Things parameters and network scanners, as well as sending commands to end devices. To configure the server, a sequence of methods *socket()*, *bind()*, *listen()* and *accept()* of the socket class object of the socket library built into the Python language is executed. These methods are called in the *launch_server()* method of the cloud platform.

After the incoming connection is accepted by the server using the *accept()* method, the method that listens for data from the connection is called.

To listen to data from a client that is a wireless network device scanner, the *client_server()* method is used. This method first saves the connection to the network scanner in the *sockets* dictionary by a key equal to the IP address of the node, and then starts a cycle of listening to data from the network scanner, which runs until the connection is terminated. The *client_server()* method decrypts and converts the data string received from the network scanner via a socket into json format, after which it enters information about the network node devices into the *client_devices* table, containing data on the IP and MAC address, type, latency and availability of devices, and also updates the values of the Internet of Things device parameters in the *device_indicators* table of the database the cloud platform of the Internet of Things.

The *client_esp32()* method is called when the client is an Internet of Things parameter collection device (ESP32 microcontroller). This method starts a cycle of listening to data from the parameter acquisition device, also running until the connection to the microcontroller is severed. After its decryption and conversion to json format, new values of the Internet of Things parameters are extracted from the string received via the socket and entered into the database.

Execution of methods for listening to data from IoT end devices is carried out in separate streams, which reduces the time to receive data from various devices, speeds up the process of updating the values of parameters in the database and the mechanism of operation of the IoT cloud platform as a whole.

To integrate with the visual software system and provide access to various data, a web server based on the REST architecture is implemented in the cloud platform. The web server allows the software system to receive data on the structure of a software-defined network of Internet of Things devices and set its configuration through the application programming interface (API). The cloud platform responds to web requests from the software system using views that have a syntax similar to functions and are mapped to one or more URL requests. The web server's *send_cloud_mac()* view returns the MAC address of the cloud platform. The *send_network_data()* view is used to obtain information about the structure of a software-defined network of Internet of Things devices, its nodes and end devices. The *send_esp32_data()* view returns the current parameter values received by a specific ESP32 microcontroller. To get a list of network configuration rules, use the *get_rules()* view. Receiving and further sending commands to control the TP-Link Tapo P100 smart plug via the socket to the network scanner, as well as returning the socket operation status, are carried out in the *manage_tapo_p100()* view. To get the network configuration rules generated by the user from the software system and save them in the database, the *add_command()* view is used.

The controller of a software-defined network of Internet of Things devices is implemented in the *controller()* method. The controller receives a list of all configuration rules generated by the user from the database and verifies their validity. If the rule is true, the controller sends the command specified in the rule to the corresponding node or end device.

Development of a visual software system for configuring Internet of Things devices

The developed visual software system for configuring Internet of Things devices, IoT Map, provides a set of tools for managing network infrastructure and Internet of Things devices, as well as visualizes the network topology and data received from connected devices.

This software system is a cross-platform application developed in the Python programming language using a library for creating a graphical user interface PyQt5. The software system is available for use on Windows and Mac OS operating systems.

Conventionally, the graphical interface can be divided into the following components: a graphical editor, a device information window, a device functions window and a program menu. The graphical editor displays the topology of a software-defined network of Internet of Things devices. The device information window contains information about the IP address, MAC address, device type, as well as the delay in transmitting commands to the device and receiving data from the device in milliseconds. The device functions window displays a list of data received from the device, a set of commands and functions supported by this device, and also allows you to configure the network.

The device configuration window allows the user to create new rules for the configuration of IoT devices.

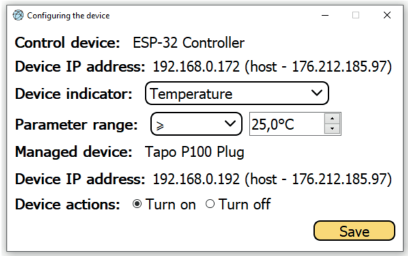
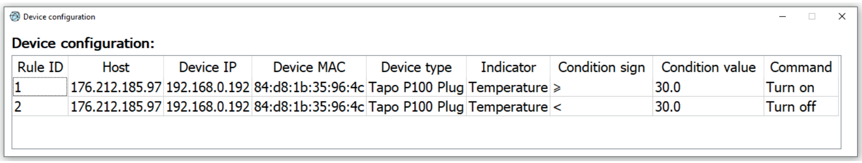


Fig. 3. Device configuration window

The device configuration display window displays information about the configuration of a specific device.

Conclusion

The article develops a cloud platform for aggregating data on the structure of the network for receiving information from end devices. To configure a software-defined network of Internet of Things devices, visualization of network topology and data received from end devices, the visual software system IoT Map was developed. The paper proposed a four-level architecture of a software-configurable network, including an additional layer of Internet of Things devices. Special attention is paid to the organization of network interaction between the end devices of the Internet of Things, visual software system and cloud platform.



Device configuration

Device configuration:

Rule ID	Host	Device IP	Device MAC	Device type	Indicator	Condition sign	Condition value	Command
1	176.212.185.97	192.168.0.192	84:d8:1b:35:96:4c	Tapo P100 Plug	Temperature	>	30.0	Turn on
2	176.212.185.97	192.168.0.192	84:d8:1b:35:96:4c	Tapo P100 Plug	Temperature	<	30.0	Turn off

Fig. 4. Device configuration display window

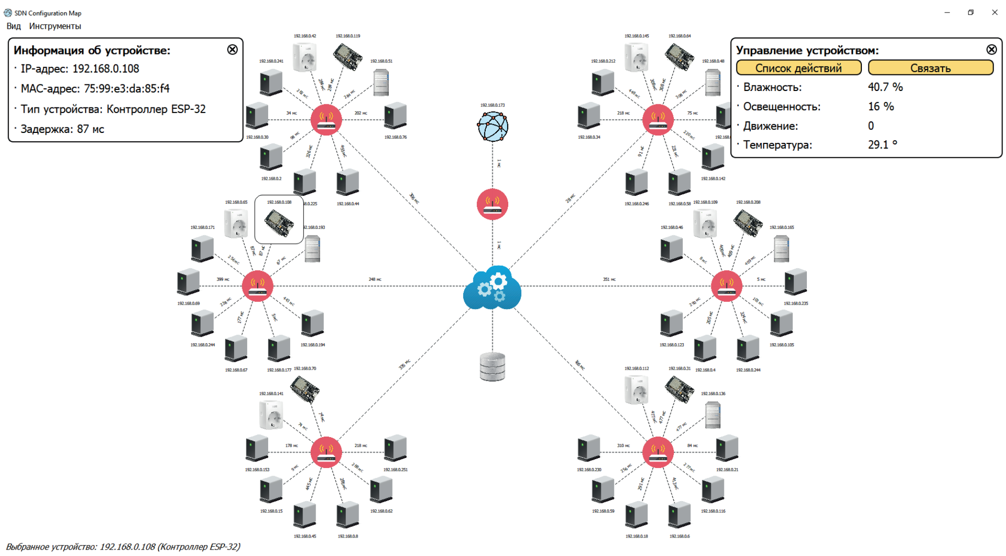


Fig. 5. Graphical interface of the IoT Map software system

References

[1] Giri A., Dutta S., Neogy S., Dahal K., Pervez Z. Internet of things (IoT): a survey on architecture, enabling technologies, applications and challenges. In: Proceedings of the 1st International Conference on Internet of Things and Machine Learning (IML'17). New York, NY, USA: Association for Computing Machinery; 2017. Article number: 7. p. 1-12. doi: https://doi.org/10.1145/3109761.3109768

[2] Ar-reyouchi E.M., Hammouti M., Maslouhi, Ghoumid H. The internet of things: network delay improvement using network coding. In: Proceedings of the Second International Conference on Internet of things, Data and Cloud Computing (ICC'17). New York, NY, USA: Association for Computing Machinery; 2017. Article number: 8. p. 1-7. doi: https://doi.org/10.1145/3018896.3018902

[3] Abid M.A., Afaqui N., Khan M.A., Akhtar M.W., Malik A.W., Munir A., Ahmad J., Shabir B. Evolution towards Smart and Software-Defined Internet of Things. *AI.* 2022;3(1):100-123. doi: https://doi.org/10.3390/ai3010007

[4] Bekri W., Jmal R., Chaari Fourati L. Internet of Things Management Based on Software Defined Networking: A Survey. *International Journal of Wireless Information Networks*. 2020;27(3):385-410. doi: https://doi.org/10.1007/s10776-020-00488-2

[5] Imran, Ghaffar Z., Alshahrani A., Fayaz M., Alghamdi A.M., Gwak J. A Topical Review on Machine Learning, Software Defined Networking, Internet of Things Applications: Research Limitations and Challenges. *Electronics*. 2021;10(8):880. doi: https://doi.org/10.3390/electronics10080880

[6] McKeown N., Anderson T., Balakrishnan H., Parulkar G., Peterson L., Rexford J., Shenker S., Turner J. OpenFlow: Enabling Innovation in Campus Networks. *ACM SIGCOMM Computer Communication Review*. 2008;38(2):69-74. doi: https://doi.org/10.1145/1355734.1355746

[7] Egilmez H.E., Gorkemli B., Tekalp A.M., Civanlar S. Scalable video streaming over OpenFlow networks: An optimization framework for QoS routing. In: 2011 18th IEEE International Conference on Image Processing. Brussels, Belgium: IEEE Computer Society; 2011. p. 2241-2244. doi: https://doi.org/10.1109/ICIP.2011.6116083

[8] Perepelkin D.A. Conceptual approach of dynamic traffic formation of software-defined telecommunication networks with load balancing. *Information Technology*. 2015;21(8):602-610. Available at: https://www.elibrary.ru/item.asp?id=24103842 (accessed 13.06.2022). (In Russ., abstract in Eng.)

[9] Koryachko V.P., Perepelkin D.A., Ivanchikova M.A., Byshov V.S., Tsyganov I.Yu. Software infrastructure and visual environment of distributed data flows processing in software-defined networks. *Vestnik of Ryazan State Radio Engineering University.* 2018;(65):44-54. (In Russ., abstract in Eng.) doi: https://doi.org/10.21667/1995-4565-2018-65-3-44-54

[10] Nikulchev E.V., Payain S.V., Pluzhnik E.V. Dynamic traffic control of cloud infrastructure with software-defined networking. *Vestnik of Ryazan State Radio Engineering University*. 2013;(45):54-57. Available at: https://www.elibrary.ru/item.asp?id=21080770 (accessed 13.06.2022). (In Russ., abstract in Eng.)

[11] Leokhin Yu.L., Fatkhulin T.D. Estimation of the possibility to provide guaranteed data rate for a client in software-configurable optical network. *Vestnik Ryazanskogo gosudarstvennogo radiotekhnicheskogo universiteta* = Vestnik of Ryazan State Radio Engineering University. 2020;(71):45-59. (In Russ., abstract in Eng.) doi: https://doi.org/10.21667/1995-4565-2018-65-3-44-54

[12] Ushakova M.V., Ushakov Yu.A. Research of virtual infrastructure network of data processing center with hybrid software-configurable switching. *Vestnik of Ryazan State Radio Engineering University*. 2021;(75):34-43. (In Russ., abstract in Eng.) doi: https://doi.org/10.21667/1995-4565-2021-75-34-43

[13] Kupriyanovsky V., Klimov A., Alenkov V., Namiot D., Sneps-Sneppe M. On the New IoT Generation – ETSI Ontology Standards and Specifications. *International Journal of Open Information Technologies*. 2019;7(9):73-81. Available at: https://www.elibrary.ru/item.asp?id=39529532 (accessed 13.06.2022). (In Russ., abstract in Eng.)

[14] Kupriyanovsky V., Sneps-Sneppe M., Namiot D., Seleznev S., Sinyagov S., Kupriyanovsky J. Web of Things and Internet of Things in the Digital Economy. *International Journal of Open Information Technologies*. 2017;5(5):38-45. Available at: https://www.elibrary.ru/item.asp?id=29226712 (accessed 13.06.2022). (In Russ., abstract in Eng.)

[15] Namiot D., Sneps-Sneppe M. On Internet of Things Programming Models. In: Vishnevskiy V., Samouylov K., Kozyrev D. (eds.). Distributed Computer and Communication Networks. DCCN 2016. Communications in Computer and Information Science. Vol. 678. Cham: Springer; 2016. p. 13-24. doi: https://doi.org/10.1007/978-3-319-51917-3_2

[16] Namiot D. [Application layer interfaces in SDN]. *Modern information technologies and IT education*. 2015;11(2):26-30. Available at: https://www.elibrary.ru/item.asp?id=26167461 (accessed 13.06.2022). (In Russ.)

[17] Magjanova A.T. [Application of cloud technologies for the implementation of Internet of Things solutions]. *Modern Innovations.* 2016;(7):30-34. Available at: https://www.elibrary.ru/item.asp?id=26397696 (accessed 13.06.2022). (In Russ.)

[18] Kupriyanovsky V., Namiot D., Drozhzhinov V., Kupriyanovsky J., Ivanov M. Internet of Things in Industrial Plants. *International Journal of Open Information Technologies*. 2016;4(12):69-78. Available at: https://www.elibrary.ru/item.asp?id=27543361 (accessed 13.06.2022). (In Russ., abstract in Eng.)

[19] Nikolaev P.L. The architecture of a smart home management system integrated into the cloud environment. *Software & Systems*. 2015;(2):65-69. (In Russ., abstract in Eng.) doi: https://doi.org/10.15827/0236-235X.110.065-069

[20] Brumshteyn Yu.M., Gayfitdinova V.Yu. The system analysis of "Smart Objects" concepts of the urban environment in the conditions of information and communication technologies development. *Caspian Journal: Management and High Technologies.* 2017;(1):24-38. Available at: https://www.elibrary.ru/item.asp?id=29156235 (accessed 13.06.2022). (In Russ., abstract in Eng.)

[21] Asaturova Yu.M. Industrial Internet in the Concept of Industry 4.0: Tasks and Development in Russia. *Economy and Entrepreneurship*. 2017;(12-3):107-113. Available at: https://www.elibrary.ru/item.asp?id=32878071 (accessed 13.06.2022). (In Russ., abstract in Eng.)

[22] Zakharov A.A., Nissenbaum O.V., Nesgovorov E.S., Ponomarev K.Yu. Constructing Authorization Model for Internet of Things. *Innovative Development of Economy*. 2017;(3):238-243. Available at: https://www.elibrary.ru/item.asp?id=29747153 (accessed 13.06.2022). (In Russ., abstract in Eng.)

[23] [20] Doceuchaev D.E., Yunusov N.T. Cloud for Internet of Things. *Social'no-èkonomičeskoe upravlenie: teoriâ i praktika*. 2018;(3):101-104. Available at: https://www.elibrary.ru/item.asp?id=35691037 (accessed 13.06.2022). (In Russ., abstract in Eng.)

[24] García-Tadeo D.A., Peram D.R., Kumar K.S., Vives L., Sharma T., Manoharan G. Comparing the impact of Internet of Things and cloud computing on organisational behavior: A survey. *Materials Today: Proceedings.* 2022;51(8):2281-2285. doi: https://doi.org/10.1016/j.matpr.2021.11.399

[25] Georgakopoulos D., Jayaraman P.P. Internet of things: from internet scale sensing to smart services. *Computing*. 2016;98(10):1041-1058. doi: https://doi.org/10.1007/s00607-016-0510-0

Submitted 13.06.2022; approved after reviewing 17.08.2022; accepted for publication 26.08.2022.

About the authors:

Dmitry A. Perepelkin, Dean of the Faculty of Computer Engineering, Ryazan State Radio Engineering University named after V.F. Utkin (59/1 Gagarina St., Ryazan 390005, Russian Federation), Dr.Sci. (Tech.), Associate Professor, **ORCID: https://orcid.org/0000-0003-4775-5745**, perepelkin.d.a@rsreu.ru

Daniil D. Tkachev, student of the Faculty Computer Engineering, Ryazan State Radio Engineering University named after V.F. Utkin (59/1 Gagarina St., Ryazan 390005, Russian Federation), **ORCID: https://orcid.org/0000-0002-1999-1356**, fenix784@mail.ru

All authors have read and approved the final manuscript.

Список использованных источников

[1] Internet of things (IoT): a survey on architecture, enabling technologies, applications and challenges / A. Giri [и др.] // Proceedings of the 1st International Conference on Internet of Things and Machine Learning (IML'17). New York, NY, USA: Association for Computing Machinery, 2017. Article number: 7. P. 1-12. doi: https://doi.org/10.1145/3109761.3109768

[2] The internet of things: network delay improvement using network coding / E. M. Ar-reyouchi [и др.] // Proceedings of the Second International Conference on Internet of things, Data and Cloud Computing (ICC'17). New York, NY, USA: Association for Computing Machinery, 2017. Article number: 8. P. 1-7. doi: https://doi.org/10.1145/3018896.3018902

[3] Evolution towards Smart and Software-Defined Internet of Things / M.A. Abid [и др.] // AI. 2022. Vol. 3, issue 1. P. 100-123. doi: https://doi.org/10.3390/ai3010007

[4] Bekri W., Jmal R., Chaari Fourati L. Internet of Things Management Based on Software Defined Networking: A Survey // International Journal of Wireless Information Networks. 2020. Vol. 27, issue 3. P. 385-410. doi: https://doi.org/10.1007/s10776-020-00488-2

[5] A Topical Review on Machine Learning, Software Defined Networking, Internet of Things Applications: Research Limitations and Challenges / Imran [и др.] // Electronics. 2021. Vol. 10, isse 8. P. 880. doi: https://doi.org/10.3390/electronics10080880

[6] OpenFlow: Enabling Innovation in Campus Networks / N. McKeown [и др.] // ACM SIGCOMM Computer Communication Review. 2008. Vol. 38, no. 2. P. 69-74. doi: https://doi.org/10.1145/1355734.1355746

[7] Scalable video streaming over OpenFlow networks: An optimization framework for QoS routing / H. E. Egilmez [и др.] // 2011 18th IEEE International Conference on Image Processing. Brussels, Belgium: IEEE Computer Society, 2011. P. 2241-2244. doi: https://doi.org/10.1109/ICIP.2011.6116083

[8] Перепелкин Д. А. Концептуальный подход динамического формирования трафика программно-конфигурируемых телекоммуникационных сетей с балансировкой нагрузки // Информационные технологии. 2015. Т. 21, № 8. С. 602-610. URL: https://www.elibrary.ru/item.asp?id=24103842 (дата обращения: 13.06.2022).

[9] Программная инфраструктура и визуальная среда распределенной обработки потоков данных в программно-конфигурируемых сетях / В. П. Корячко [и др.] // Вестник Рязанского государственного радиотехнического университета. 2018. № 65. С. 44-54. doi: https://doi.org/10.21667/1995-4565-2018-65-3-44-54

[10] Никульчев Е. В., Паяин С. В., Плужник Е. В. Динамическое управление трафиком программно-конфигурируемых сетей в облачной инфраструктуре // Вестник Рязанского государственного радиотехнического университета. 2013. № 45. С. 54-57. URL: https://www.elibrary.ru/item.asp?id=21080770 (дата обращения: 13.06.2022).

[11] Леохин Ю. Л., Фатхулин Т. Д. Оценка возможности предоставления гарантированной скорости передачи данных в программно-конфигурируемой оптической сети // Вестник Рязанского государственного радиотехнического университета. 2020. № 71. С. 45-59. doi: https://doi.org/10.21667/1995-4565-2018-65-3-44-54

[12] Ушакова М. В., Ушаков Ю. А. Исследование сети виртуальной инфраструктуры центра обработки данных с гибридной программно-конфигурируемой коммутацией // Вестник Рязанского государственного радиотехнического университета. 2021. № 75. С. 34-43. doi: https://doi.org/10.21667/1995-4565-2021-75-34-43

[13] Новое поколение интернета вещей – стандарты и спецификации онтологий ETSI / В. П. Куприяновский [и др.] // International Journal of Open Information Technologies. 2019. Т. 7, № 9. С. 73-81. URL: https://www.elibrary.ru/item.asp?id=39529532 (дата обращения: 13.06.2022).

[14] Веб Вещей и Интернет Вещей в цифровой экономике / В. П. Куприяновский [и др.] // International Journal of Open Information Technologies. 2017. Т. 5, № 5. С. 38-45. URL: https://www.elibrary.ru/item.asp?id=29226712 (дата обращения: 13.06.2022).

[15] Namiot D., Sneps-Sneppe M. On Internet of Things Programming Models // Distributed Computer and Communication Networks. DCCN 2016. Communications in Computer and Information Science ; ed. by V. Vishnevskiy, K. Samouylov, D. Kozyrev. Vol. 678. Cham: Springer, 2016. P. 13-24. doi: https://doi.org/10.1007/978-3-319-51917-3_2

[16] Намиот Д. Е. Интерфейсы прикладного уровня в SDN // Современные информационные технологии и ИТ-образование // 2015. Т. 11, № 2. С. 26-30. URL: https://www.elibrary.ru/item.asp?id=26167461 (дата обращения: 13.06.2022).

[17] Магжанова А. Т. Применение облачных технологий для реализации решений Интернета вещей // Современные инновации. 2016. № 7(9). С. 30-34. URL: https://www.elibrary.ru/item.asp?id=26397696 (дата обращения: 13.06.2022).

[18] Интернет Вещей на промышленных предприятиях / В. П. Куприяновский [и др.] // International Journal of Open Information Technologies. 2016. Т. 4, № 12. С. 69-78. URL: https://www.elibrary.ru/item.asp?id=27543361 (дата обращения: 13.06.2022).

[19] Николаев П. Л. Архитектура интегрированной в облачную среду системы управления умным домом // Программные продукты и системы. 2015. № 2. С. 65-69. doi: https://doi.org/10.15827/0236-235X.110.065-069

[20] Брумштейн Ю. М., Гайфитдинова В. Ю. Системный анализ совокупности концепций «умных объектов» городской среды в условиях развития информационно-коммуникационных технологий // Прикаспийский журнал: управление и высокие технологии. 2017. № 1(37). С. 24-38. URL: https://www.elibrary.ru/item.asp?id=29156235 (дата обращения: 13.06.2022).

[21] Асатурова Ю. М. Промышленный интернет в концепции Промышленность 4.0: задачи и перспективы развития в России // Экономика и предпринимательство. 2017. № 12-3(89). С. 107-113. URL: https://www.elibrary.ru/item.asp?id=32878071 (дата обращения: 13.06.2022).

[22] Построение модели авторизации интернета вещей / А. А. Захаров, О. В. Ниссенбаум, Е. С. Несговоров, К. Ю. Пономарев // Инновационное развитие экономики. 2017. № 3(39). С. 238-243. URL: https://www.elibrary.ru/item.asp?id=29747153 (дата обращения: 13.06.2022).

[23] [20] Докучаев Д. Е., Юнусов Н. Т. Облачные технологии для «Интернет вещей» // Социально-экономическое управление: теория и практика. 2018. № 3(34). С. 101-104. URL: https://www.elibrary.ru/item.asp?id=35691037 (дата обращения: 13.06.2022).

[24] Comparing the impact of Internet of Things and cloud computing on organisational behavior: A survey / D. A. García-Tadeo [и др.] // Materials Today: Proceedings. 2022. Vol. 51, Part 8. P. 2281-2285. doi: https://doi.org/10.1016/j.matpr.2021.11.399

[25] Georgakopoulos D., Jayaraman P. P. Internet of things: from internet scale sensing to smart services // Computing. 2016. Vol. 98, issue 10. P. 1041-1058. doi: https://doi.org/10.1007/s00607-016-0510-0

Поступила 13.06.2022; одобрена после рецензирования 17.08.2022; принята к публикации 26.08.2022.

Об авторах:

Перепелкин Дмитрий Александрович, декан факультета вычислительной техники, ФГБОУ ВО «Рязанский государственный радиотехнический университет им. В.Ф. Уткина» (390005, Российская Федерация, г. Рязань, ул. Гагарина, д. 59/1), доктор технических наук, доцент, **ORCID: https://orcid.org/0000-0003-4775-5745**, perepelkin.d.a@rsreu.ru

Ткачев Даниил Дмитриевич, студент факультета вычислительной техники, ФГБОУ ВО «Рязанский государственный радиотехнический университет им. В.Ф. Уткина» (390005, Российская Федерация, г. Рязань, ул. Гагарина, д. 59/1), **ORCID: https://orcid.org/0000-0002-1999-1356**, fenix784@mail.ru

Все авторы прочитали и одобрили окончательный вариант рукописи.