Selection of Protocols for Integration of Sensory Data Networks in Cloud Structures

https://doi.org/10.3991/ijoe.v18i09.31321

Filip Tsvetanov(✉), Martin Pandurski
South-West University, Blagoevgrad, Bulgaria
ftsvetanov@swu.bg

Abstract—The striving to obtain more detailed information about the environment and control various processes leads to an increase in the number of connected sensor devices in various industrial areas. The collected large amount of data can be analysed in real-time. The sensors that build up the WSN have limited hardware resources and cannot process large amounts of data. The integration between WSN and cloud structures is an excellent method for storing, processing, accessing data via the Internet and solves the issue of the limited capacity of WSN. The big challenge to designing the WSN—cloud systems is establishing a communication channel (through different protocols) between devices in the network and cloud platforms. This project executes/perform a real experiment on the XBee sensor network and the ThingSpeak cloud, and the data transmission between them is forwarded using different protocols (HTTP, HTTPS, MQTT and MQTT-SN). The influence of the parameters of the transmitted packet on the delay, the CPU, RAM load has been studied. The results give some advantages of MQTT over other protocols in terms of data rate, CPU and RAM load when working with XBee sensor modules and integration between WSN and cloud structures.

Keywords—sensor networks, HTTP, HTTPS, MQTT and MQTT-SN, clouds, integrations

1 Introduction

WSNs are the technological "backbone" for implementing the IoT paradigm in many areas of our modern life. The scope of their applications includes environmental monitoring, intelligent networks, industrial automation, etc. [1], [2], [3]. They increase the ability to share and analyse large amounts of sensor data in real-time. At the same time, sensor nodes are miniature stand-alone devices with limited resources, capable of collecting and processing data from the surrounding area and transmitting this data via wireless radio transmitters over short distances.

Scalable, high-performance computing infrastructure is needed to effectively use the amount of sensor data networks for real-time data processing/storage. The variety of cloud platforms represents such infrastructures. The cloud is a suitable method for storing, processing, accessing data over the Internet. Usually, WSNs connect to the Internet through a gateway, interact with cloud structures, providing a range of services and information, allowing remote access to devices via secure communication channels in real-time. This large amount of data is a prerequisite for many companies to use cloud databases for storage/processing sensor data [4], [5], [6], [7].

Modern requirements for real-time access and analysis of sensory data make their integration into the clouds increasingly necessary [8], [9], [10]. According to the Eclipse Foundation, concerns about the complexity of integration are also growing from 22% in 2020 to 27%. As the number of deployments increases, the need for additional integrations with complementary technologies and systems becomes apparent [11].

The biggest challenge in designing IoT systems is establishing a communication channel between local devices, gateways and cloud platforms. Data can be exchanged under various protocols [12], [13]. The complete communication stack containing the protocols is shown in Table 1.

Table 1. IoT communication protocol stack

Layers	Protocols
Application layer	HTTPS, HTTP, REST, RESTful, *MQTT*, CoAP, LWM2M (Lightweight M2M)
Transport layer	TCP, UDP
Internet layer	Two versions of IP are used: version 4 (IPv4) and version 6 (IPv6).
Link-layer	IEEE 802.15.4, Z wave, 802.11 WiFi, Bluetooth Low Energy (LE), Zigbee, NFC, GPRS/2G/3G/4G/5G, Ethernet, FID, Sigfox

The choice of the appropriate protocol depends on the conditions (applications) in which it is used (type of hardware/network, the format of the transmitted data, etc.). There are comparisons between protocols in the literature, but comparisons have been made from a technical point of view [14]. So, further analysis of each protocol is needed on how each protocol would fit into the integrated sensor networks, gateways and clouds [15], [16], [17].

The communication between the WSNs and the cloud computing system can be realised in different scenarios depending on data transmission through a gateway or coordinator or directly between the sensor network and the cloud [18]. An example of such an architecture is the one proposed in [19] of the WSN—cloud system. The data was collected through the Raspberry Pi gateway from sensors measuring the patient's physical parameters during surgery. An interesting proposal is the possibility to send a voice signal to the surgeon in case of need for help from other surgeons monitoring the online operation. It is possible through Raspberry pi cameras through the server. Competent specialists can give instructions and communicate with the operator in the operating room through headphones and a microphone.

Research related to the integration of WSNs into cloud structures is an attractive and topical issue that attracts the attention of both the scientific community and the industry. The relevance of the work is determined by the fact that the integration of a cloud network provides opportunities to develop innovative and new interesting approaches for data collection, processing, remote monitoring and management in many areas of idols and critical infrastructure. The ability to research and analyse the operation of networks before their physical construction gives a definite competitive advantage for their application in various industrial areas.

Another important problem that arises due to the integration between WSN and the cloud is the security of the data transmitted over the communication channel. A shared responsibility model allows cloud providers to ensure that the hardware and software services they provide are secure while cloud users remain responsible for the security of their data assets. Cloud providers often offer better security than many companies can achieve independently. In work [20], the security of WSN was studied by introducing a trust model based on cloud theory. The results show that the trust model has good resilience by accurately identifying malicious nodes and preventing network destruction. The authors conclude that the idea of cloud theory can effectively improve network security.

The problem with the security of the transmitted data focuses on researchers and industry. Various measures are being taken, including designing specialised controllers, developing specialised hardware protection, and implementing appropriate protocols to integrate the two parts of the system.

The present work aims to study the influence of the transmitted data parameters of popular integration protocols HTTP, HTTPS, MQTT, MQTT_SN between sensor or IoT network to cloud on the transmission packet delay CPU/RAM load. The results of the experimental study give some advantages of the MQTT [22] over other protocols in terms of data rate and CPU/RAM load when working with XBee sensor devices.

2 Material and methods

2.1 Description of the experimental study

By estimating the connection from sensor data networks to the clouds, real-time data can be analysed, allowing the management of certain conditions. Different criteria can do the integration assessment, for example, support applications that work with varying communication models and provide different data speed integration. The general requirement is the reliable transmission of sensor data to the cloud's database. In the study, as a target function, determining the quality of integration, we take the delay of data transmissions (speed) and CPU/RAM load, showing the energy consumption.

The research is carried out with a real-built XBee sensor network, which reads real-time data such as temperature/light—the XBee modules are used with gateways to establish Xbee end-to-end wireless network connection, Figure 1.

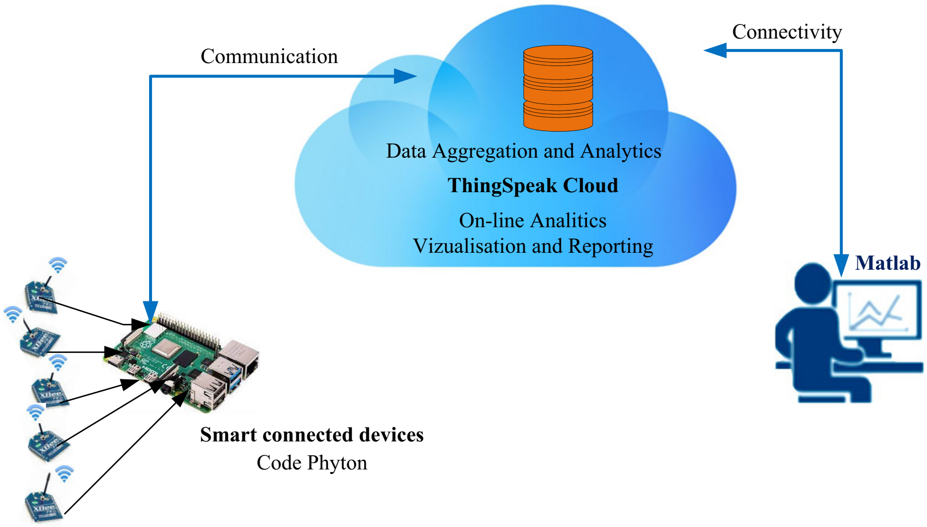

Communication
Connectivity
Data Aggregation and Analytics
ThingSpeak Cloud
On-line Analitics
Vizualisation and Reporting
Matlab
Smart connected devices
Code Phyton

Fig. 1. Sensor data integration into ThingSpeak

The data generated from sensors is transmitted to a gateway by connecting the XBee modules and integrated into a cloud (broker/server); via several different protocols: HTTP, HTTPS, MQTT and MQTT-SN. For this purpose, a Python program has been developed that sends predefined messages organised in cycles. The program defines several parameters, such as the number of transmitted data packets, the number of topics per packet and the byte value per topic. The main experiments measure data delay and CPU/RAM load while the protocols send data to the cloud. The collected sensor data are processed, stored and analysed in the ThingSpeak.

2.2 Architecture of the experimental network

The collected XBee sensor data is forwarded to RPI4, which loads the pre-developed Python code for the experiment (Figure 2). RPI4 forward sensor data to the ThingSpeak Cloud [19] via MQTT, HTTP, HTTPS and MQTT-SN. ThingSpeak is code-free and supports various integration protocols. ThingSpeak is not designed to access MQTT-SN data. It requires using MQTT-SN Gateway, which converts MQTT-SN messages into MQTT (UDP) messages. The Eclipse Paho MQTT-SN client library is used to perform this task, which is realised through a process of serialisation and deserialisation [21].

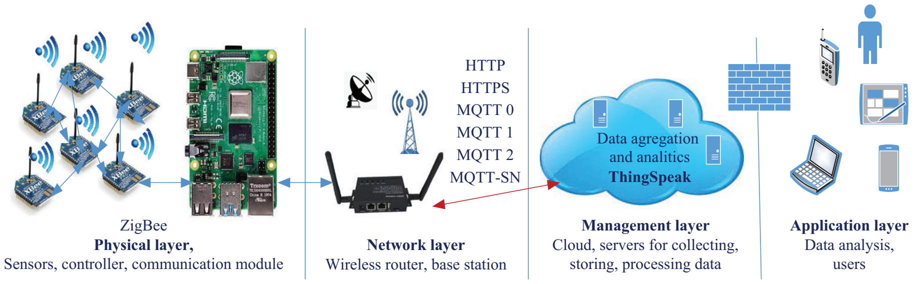


Fig. 2. The architecture of the experimental network

2.3 Variable parameters

The variable parameters for the conducted experiment for each protocol are the number of transmitted packets, number of topics per packet and byte value per topic. The objective is to test how data transmission through the protocols will affect the transmission data delay, forwarding to the cloud. The parameters values are measurable and are determined when setting up experiments.

2.4 Algorithm for conducting the experiment

First step: Setting the sensor nodes in "broadcast" mode for data transmission in the mesh network topology to RPI4.

Second step: Development of a code for testing the operability of the protocols.

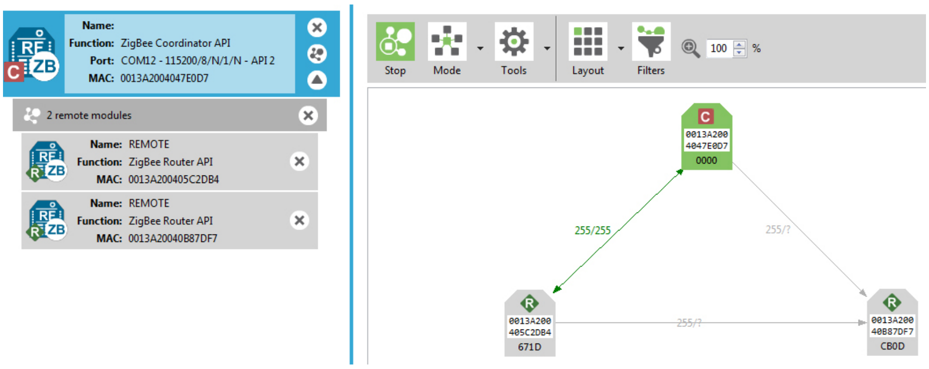
Third step: Setting up the parameters for conducting the study in different scenarios: number of transmitted packets, number of topics per packet and byte value per topic.

Fourth step: Connect to the cloud, generate and process data, and access the storage database.

2.5 Experiment settings

1) Settings of the Xbee modules

We set the Xbee Coordinator to remotely receive sensor data from other Xbee Routers by configuring them to work in "broadcast" mode for data transmission in the mesh network topology to RPI4 (Figure 3).



Name:
Function: ZigBee Coordinator API
Port: COM12 - 115200/8/N/1/N - API 2
MAC: 0013A2004047E0D7

2 remote modules

Name: REMOTE
Function: ZigBee Router API
MAC: 0013A200405C2DB4

Name: REMOTE
Function: ZigBee Router API
MAC: 0013A20040B87DF7

Fig. 3. Xbee mesh topology

2) Registration and settings in ThingSpeak

To run the experiment, we need registration in ThingSpeak (Figure 4), which works with HTTP and MQTT. Creating a virtual channel (with fields and access point API key) will direct the messages from sensory data to the field [23].

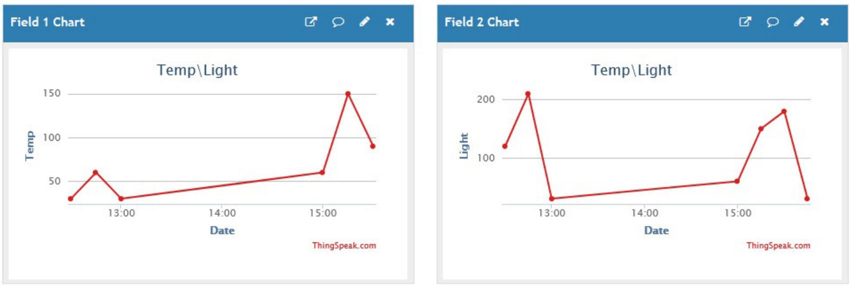


Fig. 4. ThingSpeak integrated data

3) Configuration of protocols for integration and implementation of HTTP and MQTT clients

The properties of MQTT and HTTP affect how the tests are performed. MQTT supports the publishing/subscribing model, in which clients connect to a broker, and remote devices publish messages in a shared queue. HTTP supports the request/response model. The generated code is an HTTP client that sends data packets to the cloud, while for MQTT (TCP) settings, we use an additional Eclipse Paho MQTT client packet. The Eclipse Paho MQTT-SN Gateway is also used for MQTT-SN, which converts MQTT-SN (UDP) to MQTT (UDP).

3 Results and discussion

a. *Investigation/survey of the byte value influence of the transmitted data packets on the delay.*

The experiment is performed with 200 packets and 200 data points per packet.

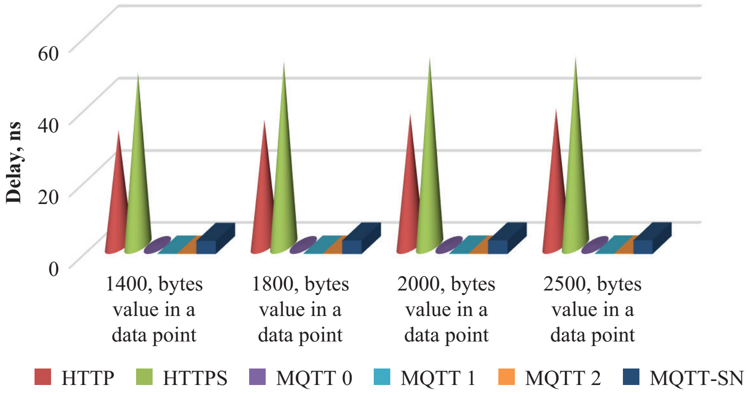


Fig. 5. Influence of byte value on the delay at 200 packets, 200 data points in packets, ns

The most significant delay is with HTTPS due to the provision of TLC protection, and the smallest is in MQTT due to the smaller topic size (Figure 5).

b. *Investigation of the byte value influence of the transmitted packets on CPU load.*

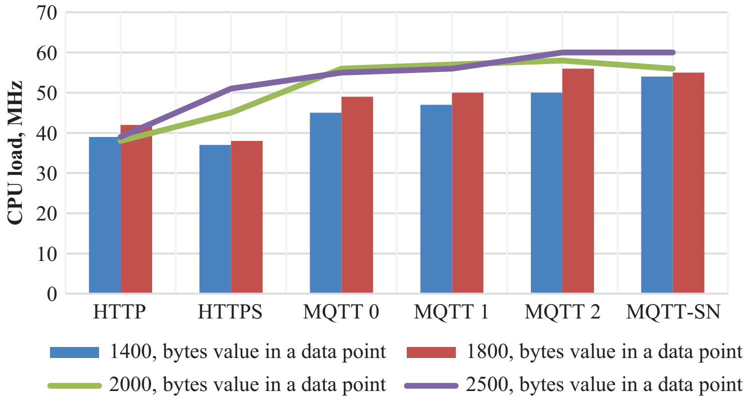


Fig. 6. The influence of the byte value of the packets at 200 packets, 200 data points on the average CPU load

The increased byte value of the topics affects CPU load most in MQTT-2 (more data transfer attempts are required) than in MQTT-0 and -1, Figure 6. The second place is CPU load level on MQTT-SN due to additional data transformations. The CPU load on HTTP is the lowest in this particular sample.

c. *Investigation of byte value influence of topics per package on RAM load.*

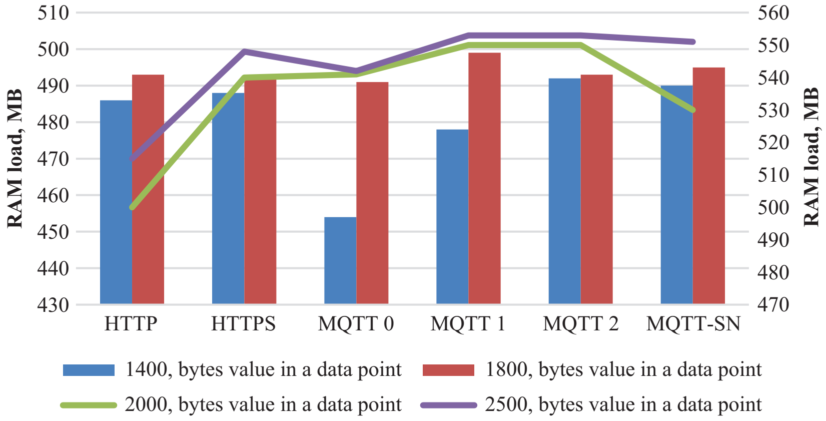


Fig. 7. The influence of the byte value of the packets at 200 packets, 200 data points on the RAM load

Due to the described features, the RAM load is highest at MQTT-SN and lowest at MQTT-0 (Figure 7).

d. *Investigation of the topic's number influence the delay in transmitted data packets.*

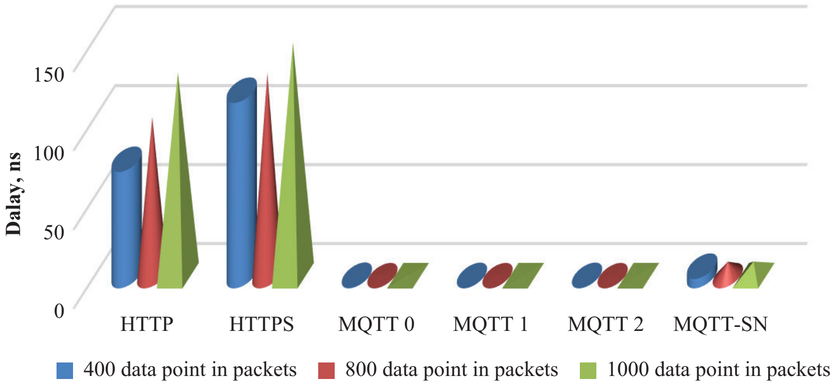


Fig. 8. The change in delay in 200 packets, 100 bytes value in a data point, depending on the data point in a packet, ns

Increasing the number of topics per packet makes the delay most significant in HTTPS. And the number of topics in the MQTT- 0,1,2 has a negligible effect on the delay.

e. *Investigation of the topics number influence in the packages on RAM load.*

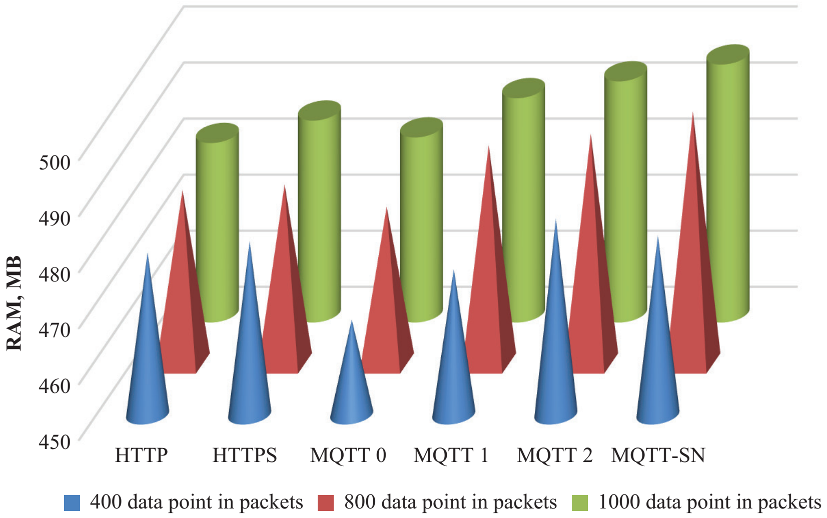


Fig. 9. Influence of the change in the number of data points in packets on RAM at 200 data packets, 100 bytes value in a data point MB

The results show in Figure 8 that RAM load is higher for the MQTT-SN, HTTPS and MQTT-2 (due to the described features) and the lowest for MQTT-0. The increased number of topics in one package increases the number of required operations and time processing, which leads to a higher RAM load (Figure 9).

f. *Investigating the topics number influence in the packages on CPU load.*

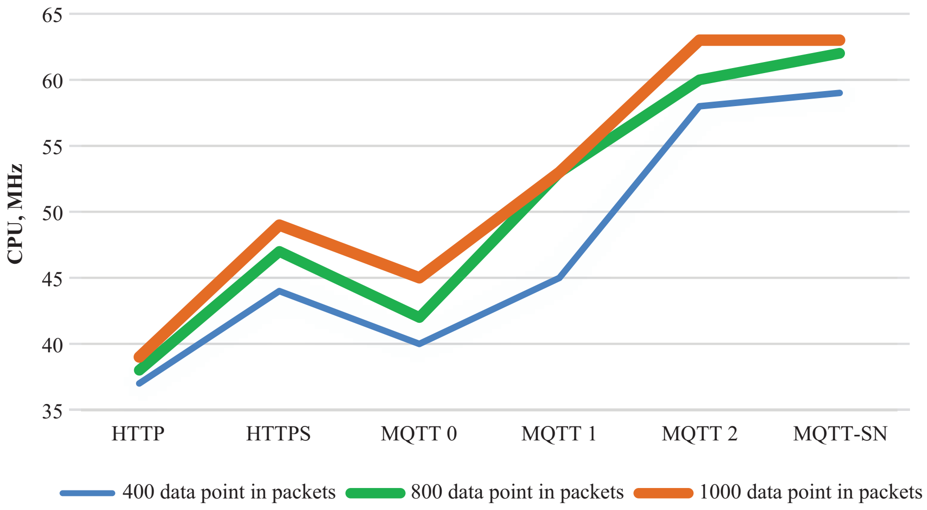


Fig. 10. The influence of the number of the topics in the packages on the CPU load

CPU load is most significant with MQTT-SN (Figure 10), as MQTT-SN uses more complex data conversion. It is also interesting that HTTPS also has a higher load due to the additional use of TLC. Practically the smallest CPU load is at MQTT-0.

4 Conclusion

The integration between sensor networks and cloud structures allows solving the problems related to the hardware limitations of the sensor networks for storage and processing of the collected data without unnecessarily increasing the cost of the sensor networks. One of the biggest challenges in designing integrated cloud-sensor network systems is establishing a communication channel (via different protocols) between devices, gateways, and cloud platforms. The paper presents the results of a study in which we choose the best integration protocol exclusively for sensor data. The parameters of the transmitted data packets such as delay, CPU and RAM load were studied. The results give MQTT advantages over other protocols regarding data speed, CPU and RAM load when working with XBee sensor devices. These results confirm the findings of the Eclipse Foundation study [11], according to which MQTT is preferred mainly by the industry for the integration of sensor data with cloud structures. The advantage of our research is that we confirm this conclusion by building a real sensor network with specific quantitative results for the delay of data transmission and CPU and RAM load.

5 References

[1] M. N. Pandurski and F. A. Tsvetanov, "Application of Sensor Networks for Measuring Insulin Levels", *International Journal of Online and Biomedical Engineering (iJOE)*, 16(14), pp. 122–136, 2020. https://doi.org/10.3991/ijoe.v16i14.17185

[2] D. Radha, M. Kumar, N. Telagam, and M. Sabarimuthu, "Smart Sensor Network-Based Autonomous Fire Extinguish Robot Using IoT", *International Journal of Online and Biomedical Engineering (iJOE)*, 17(1), pp. 101–110, 2021. https://doi.org/10.3991/ijoe.v17i01.19209

[3] A. Lanzolla and M. Spadavecchia, "Wireless Sensor Networks for Environmental Monitoring", *Sensors* (Basel, Switzerland), 21(4), p. 1172, 2021. https://doi.org/10.3390/s21041172

[4] B. Salmon, "Understanding Cloud Storage Models," https://www.infoworld.come/2871290/understanding-cloud-storage-models.html [Accessed January 18, 2022].

[5] F. A. Tsvetanov, "Storing Data from Sensors networks", *Journal IOP Conference Series: Materials Science and Engineering*, 1032, p. 012012, 2021, https://doi.org/10.1088/1757-899X/1032/1/012012

[6] A. Eisa, E. H. M. El-Bakry, S. M. Abd Elrazik, S. Q. Hasan, A. Q. Hasan, and S. Zaid, "Challenges in Wireless Sensor Networks", *International Journal of Advanced Research in Computer Science & Technology*, 4, pp. 22–27, 2016.

[7] M. A. Imran, A. Zoha, L. Zhang, and Q. H. Abbasi, "Grand Challenges in IoT and Sensor Networks", *Frontiers in Communications and Networks*, 1, p. 619452, 2020, https://doi.org/10.3389/frcmn.2020.619452

[8] U. H. Fischer, S. Hoppstock, P. Kußmann, and I. Steuding, "Real-Time Capable Sensor Data Analysis-Framework for Intelligent Assistance Systems", Data Acquisition – Recent Advances and Applications in Biomedical Engineering, London, United Kingdom: *IntechOpen*, 2021. https://doi.org/10.5772/intechopen.93735

[9] K. G. Shanthi, S. Sesha Vidhya, G. Vishali, R. V. Uma, M. Thanuja, and S. Srinitha, "Industrial Wireless Sensor Networks With Real Time Data Access", *Materials Today: Proceedings*, 2021. https://doi.org/10.1016/j.matpr.2021.07.033

[10] P. Patel, D. Dave, K. H. Solanki, and K. J. Modi, Improving Cloud Integrated Sensor Network Architecture, Applications & Challenges. 5th International Conference on Computing Methodologies and Communication (ICCMC), pp. 102–108, 2021. https://doi.org/10.1109/ICCMC51019.2021.9418370

[11] IoT & Edge Developer Survey Report Eclipse Foundation, https://f.hubspotusercontent10.net/hubfs/5413615/IoT%20&%20Edge%20Developer%20Survey%20Report%20-%202021.pdf?hsCtaTracking=f9d1c47d-0bd0-4180-aa7e-02f9268c73f9%7C4b9ca4f2-fa0d-4f59-8c10-a76479687f7d [Accessed January 30, 2022].

[12] H. Liazid and M. Lehsaini, "A Brief Review on Integration Between Wireless Sensor Networks and Cloud", *Concurrency Computat Pract Exper*, 33, p. e6328, 2021. https://doi.org/10.1002/cpe.6328

[13] A. Bhawiyuga, D. P. Kartikasari, K. Amron, O. Bagus Pratama, and M. W. Habibi, "Architectural Design of IoT-cloud Computing Integration Platform", *Telkomnika*, 17(3), pp. 1399–1408, 2019. https://doi.org/10.12928/telkomnika.v17i3.11786

[14] M. Burhan, R. Rehman, B. Khan, and B. Kim, "IoT Elements, Layered Architectures and Security Issues: A Comprehensive Survey", *Sensors*, 18(9), pp. 1–34, 2018. https://doi.org/10.3390/s18092796

[15] D. Soni and A. Makwana, "A Survey on MQTT: a Protocol of Internet of Things (IoT)", *International Conference on Telecommunication, Power Analysis and Computing Techniques*, 2017.

[16] M. ObulaReddy and J. B. Seventline, "Performance Analysis of QoS for the MQTT-SN Protocol with Industry Oriented Gateway and Integration with Cloud MQTT-Server, IOT-Application", *International Journal of Future Generation Communication and Networking*, 13(3), pp. 2651–2673, 2020.

[17] HTTP vs MQTT performance tests, Comparison of HTTP and MQTT protocols for performance, power consumption, and CPU usage in different scenarios on a laptop and Raspberry Pi., https://flespi.com/blog/http-vs-MQTT-performance-tests [Accessed Junuary12, 2022].

[18] F. Tsvetanov and M. Pandurski, "Some Aspects for the Integration of Sensor Networks in Cloud Structures", *Proceedings of International Conference on High Technology for Sustainable Development HiTech 2018*, pp. 249–253, 2018. https://doi.org/10.1109/HiTech.2018.8566509

[19] R. Shi and W. Xi, "Security Optimization of Wireless Sensor Networks Based on Cloud Platform", *International Journal of Online and Biomedical Engineering (iJOE)*, 14(2), pp. 48–59, 2018. https://doi.org/10.3991/ijoe.v14i02.8201

[20] R. M. T. Yanni, H. M. El-Bakry, A. Riad, and N. El-Khamisy, "Internet of Things For Surgery Process Using Raspberry Pi", *International Journal of Online and Biomedical Engineering (iJOE*), 16(10), pp. 96–115, 2020. https://doi.org/10.3991/ijoe.v16i10.15553

[21] ThingsBoard Cloud Documentation https://thingsboard.io/docs/paas/ [Accessed Junuary 22, 2022].

[22] A. Stanford-Clark and H. Linh Truong, "MQTT For Sensor Networks (MQTT-SN) Protocol Specification Version 1.2", November 14, 2013, https://www.oasis-open.org/committees/download.php/66091/MQTT-SN_spec_v1.2.pdf [Accessed October 23, 2021].

[23] ThingsBoard, Open-source IoT Platform, Device Management, Data Collection, Processing and Visualisation for Your IoT Solution, https://thingsboard.io/docs/paas/user-guide/integrations/ttn/ [Accessed 15 October 2021].

6 Authors

Filip Tsvetanov is Chief Assistant Professor in the Department of Communication and Computer Engineering and Technologies, Faculty of Engineering, in South-West University “Neofit Rilski”, Blagoevgrad, Bulgaria. PhD in Communication Engineering. Scientific Interest: communication problems in wireless sensor networks, computer networks, the integration between sensor networks and cloud structures.

Martin Pandurski is a PhD student in the Department of Communication and Computer Engineering and Technologies in the Faculty of Engineering at South-West University, Blagoevgrad, Bulgaria. The object of his research is sensor networks, the integration between sensor networks and cloud structures. E-mail: tom1000@abv.bg

Article submitted 2022-03-30. Resubmitted 2022-05-13. Final acceptance 2022-05-13. Final version published as submitted by the authors.