Journal of Kyzylorda Scholarly Review

P-ISSN:3006-0265 | E-ISSN:3007-4681. Published by the Open University of Kyzylorda

https://bulletin.ouk.kz/index.php/bulletin/index

Evaluating Reliability of Communication in IoT based Water Monitoring Using ESP32

Mustafa Mohammed Jasim[1,*], Firas Mohammed Adress[2], Ayat Abbas Fadhil[3]

[1,2] Nineveh Education Directorate, Mosul, Iraq.
[3] Department of Artificial Intelligence, College of Science, University of Alkafeel, Najaf, Iraq
[*] Corresponding Author: tecmustafa90@gmail.com

Abstract. In IoT-based technology solutions, there exist several tools for establishing connection with sensors and maintaining the flow of data between them. One of the commonly used solutions in IoT systems for sending and receiving data between connected devices is the Message Queuing Telemetry Transport (MQTT). This communication protocol features a lightweight data transfer mechanism and can be used by the devices with low processing power and memory. In this paper, we propose an experimental model of an IoT-built water monitoring network based on ESP32 module to conduct an evaluation of reliability and communication properties of the MQTT communication protocol at different transmission rates. In this system, an ultrasonic sensor is incorporated into the ESP32-based device for collecting information about the level of water and its subsequent transfer via the MQTT communication protocol. The evaluation of the system will be conducted based on several parameters, such as average delay, availability, average throughput, average packet loss ratio (PLR), average packet delivery ratio (PDR), and sensor reliability. Several transmission rates will be considered in order to evaluate their influence on communication process and system reliability. According to the experimental results, higher transmission rates increase throughput but decrease latency, packet delivery, availability, packet loss, and accuracy of measurements. At the lowest transmission rate, the system achieves 98% availability and maintains packet delivery with low packet loss.

Keywords: Internet of Things (IoT) Communication, Latency, Throughput, Packet loss, and Message Queuing Telemetry Transport (MQTT) Protocol.

1. INTRODUCTION

Technology known as Internet of Things (IoT) has become an important technology of communication for interconnected sensors network, thus allowing different devices to communicate. IoT offers a medium for various applications such as smart houses, smart transport, agriculture, and automated systems. With an increase in the number of devices that are becoming interconnected through IoT platforms, IoT applications need communication technologies that will allow them to maintain constant connectivity while at the same time providing low average delay and acceptable average throughput and availability. Messages Queueing Telemetries Transports (MQTT) which is a type of communication protocol that has been specifically developed for the purpose of communication in the IoT environment. It follows the model of publish/subscribe where a publisher sends information to the broker and subscribers receive the information from the broker. This model of communication simplifies the communication process and allows information to be exchanged efficiently among the distributed devices [3]. Moreover, MQTT uses a relatively lightweight communication structure and needs relatively low amounts of computational and networking resources making it a good protocol for communicating in environments that have constrained resources [4]. It provides different types of communication options such as one-to-one and many-to-many communication [5]. Previous investigations have examined MQTT from several perspectives, particularly in relation to broker performance, communication efficiency, reliability, security, and interoperability. A comparative evaluation of MQTT brokers reported that Mosquitto provided favorable results regarding latency and CPU utilization when compared with Bevywise, ActiveMQ, HiveMQ, and EMQX [6]. Another study introduced a systematic approach for classifying and assessing IoT brokers, combining a conceptual framework with middleware development and practical implementation for smart environments [7]. The paper also stresses the importance of improving the performance of brokers and developing the means for achieving interoperability between different broker architectures. Besides, security-oriented literature has paid attention to the MQTT-based architectures and utilized broker collaboration and interest-based data delivery to increase the optimization and protection of connection into IoTs systems [8]. Also, a publish/subscribe architecture based on MQTT has been experimentally tested and showed reliability of more than 96%, as well as communication delays less than 300 ms [9]. Comparative studies between MQTT and CoAP have explored the communication features of the two protocols and found MQTT more appropriate for IoT scenarios with several devices interconnected [10]. Moreover, blockchain technology has been integrated into the IoT sensor systems based on MQTT to develop forensics and protect from digital attacks [11]. MQTT has also been used in the communication architecture of smart-factories to provide the interaction between equipment using heterogeneous industrial communication technologies [12]. In addition, performance studies of MQTT have included critical issues, such as communication delay and energy consumption, and found MQTT having a lower delay compared to QUIC [13]. Comparative research on real-time IoT communication protocols has included MQTT together with RTPS, HTTP, and AMQP and found that in case of message delivery MQTT may be more advantageous compared to others due to short messages [14], [15]. Finally, MQTT-based load-testing tools have been developed to explore the behavior of the system under intensive IoT data traffic and are implemented for graphic and text-based operating systems [16]. While all the research works above have revealed the applicability of MQTT in a wide range of IoT application contexts, a detailed evaluation of the reliability of communication at different data transmission rates still needs to be conducted, especially for water-level monitoring applications. Therefore, in this paper, an IoT water monitoring prototype based on ESP32 is developed, and the experiment concerning the communication reliability of the device with the use of MQTT is conducted. The assessment will be performed according to such criteria as average delay, availability, average throughput, average PLR, average PDR, and sensor reliability at different data transfer ratios. This work seeks to determine the impact of different transmission rates on the communication process and to identify the optimal conditions for communication in the case of water-level monitoring. The main novelty of this research is that it provides an integrated experimental study of MQTT communication in ESP32-based system, paying special attention to the effect of the transmission rate on several communication reliability criteria. The rest of this paper is structured in the following manner. The second section explains the ESP32-powered water monitoring system and the chosen evaluation process. The results of our experiments are reported in the third section of the work. Finally, Section 4 offers conclusions from this research.

2.METHODOLOGY

2.1. Monitor System Design

The designed IoT-based water monitoring system seeks to establish the effectiveness of data transfer under different conditions of data transfer. Both hardware and software elements have been used to implement the prototype as well as analyze its communication capability. The complete system architecture has been designed using an ultrasonic sensor, an ESP32-based NodeMCU board, MQTT broker, and a Node-RED program. In this setup, the ESP32-based board acts as the publisher whereas the Node-RED acts as the subscriber. The MQTT broker offers the middle level for communication that receives and forwards the data from the publisher to the subscriber.

An ultrasonic sensor has been embedded within the prototype for data collection about the water level. These types of sensors offer an economical approach to sensing, and they work under various conditions, which may include changes in pressure and temperature [17]. While working, the sensor measures the level of the liquid, and the measurement information is transmit to the ESP32 NodeMCU. The ESP32 is considered a MCUSoC platform because it incorporates wireless communication capabilities like Wi-Fi and Bluetooth [18]. Cost effectiveness, energy efficiency, and computational features are some of the reasons why the ESP32 has been adopted in IoT projects [19]. Figure 2 illustrates the physical setup between the ultrasonic sensor and ESP32 NodeMCU. For communicating purposes, the ESP32 NodeMCU board is attached to the Wi-Fi modem. When a wireless connection is established, data obtained from the ultrasonic sensor is sent from the ESP32 to the MQTT broker using the network. Then, the MQTT broker organizes the messages published and delivers them to the subscribing application. In such a way, communication is enabled that allows transmitting the measurement results from the sensor to the monitoring interface via the publish-subscribe protocol. The Node-RED software acts as the subscribing application to build up the monitoring interface. It is a visual programming environment that facilitates connections between equipment hardware, online platforms, and software development connections, while also supporting application development and workflow automation [20]–[22]. Its capabilities make it suitable for IoT applications involving device communication, sensor-data collection, processing, and monitoring.

Figure 3 clearly depicts that the entire process starts when the ultrasonic sensor measures the level of the liquid in a water tank. The measurements are then delivered to the ESP32 Node-MCU, which publishes the generated data through MQTT mediator. The mediator transfers messages into Node-RED subscriber, where received information can be monitored and analyzed. This architecture defines the channel used for communication in evaluating the chosen performance indicators of the IoT-based water monitoring system.

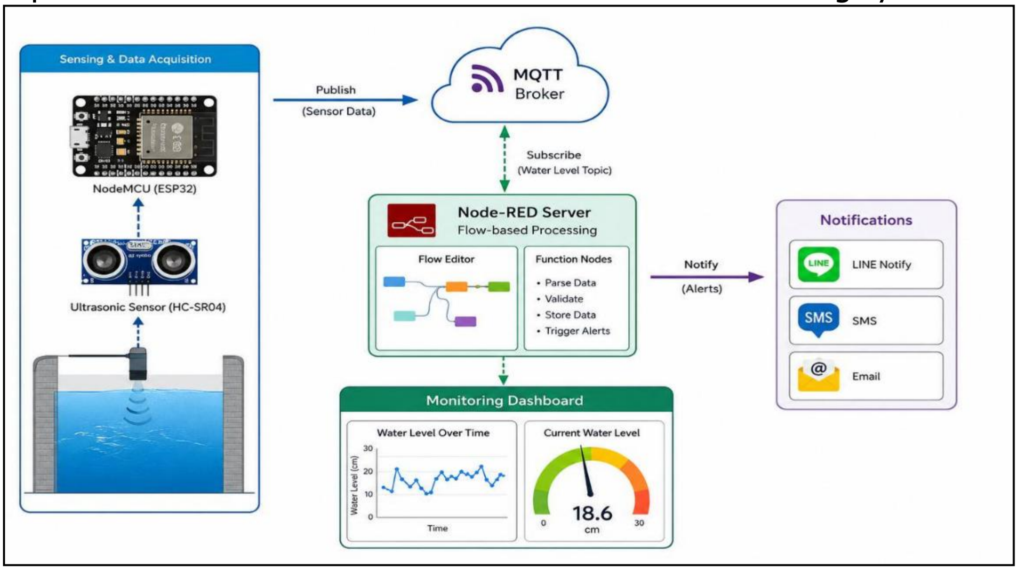

Figure 1. Overall Architecture about the planned IoT-Based Level surveillance structure

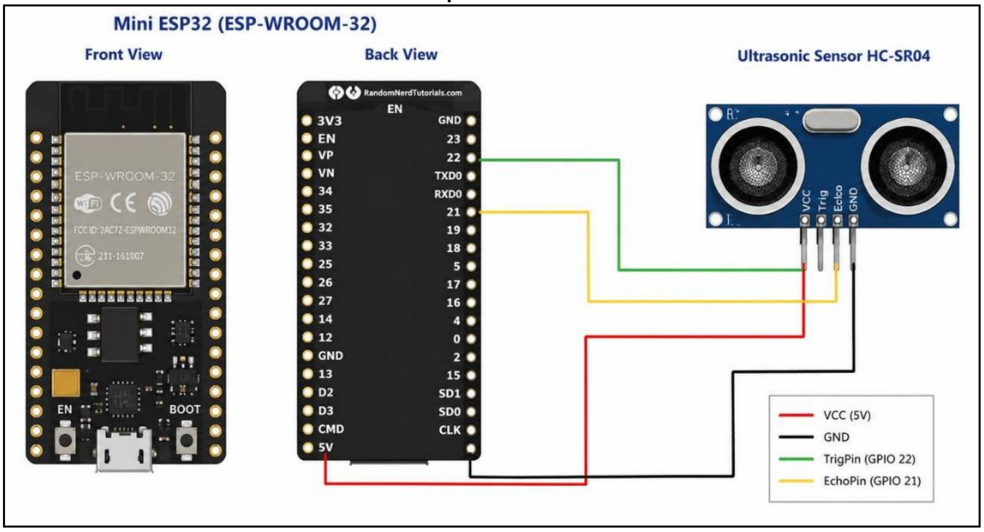

Figure 2. Hardware Interface between the Mini ESP32 and HC-SR04 Ultrasonic Sensor

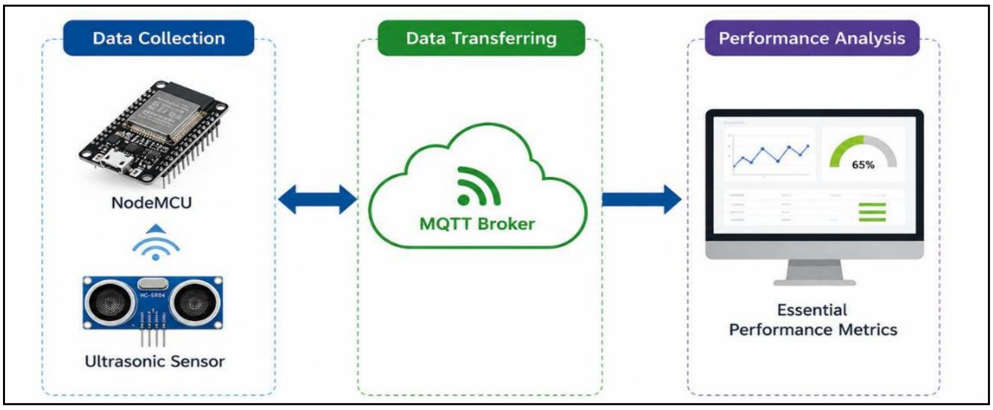

Figure 3. IoT-Based Data Collection, Transmission, and performance Analysis Architecture

2.2. Parameters for performance evaluation

The proposed system is tested with a set of communication and sensing parameters in order to describe its performance characteristics. The six factors that have been chosen for investigation include average delay, average throughput, availability, average PLR, average

PDR, and sensor reliability. This set of performance parameters is used to estimate effectiveness of the process of data transmission and the accuracy of the water level measuring carried out by the monitoring system. The definitions and computation procedures of each parameter are presented below.

2.2.1. End to End Delay (Latency)

Latency is the time period connected with delivery of data across the communication channel and is one of the performance measures of the system under investigation. For the proposed system, latency implies the time that takes place during transmitting the data packet from producer to the follower. Low average delay means quick delivery from sensor data via the communication channel. In this research, the latency is computed depending on the relation between packet size and bandwidth used for transmitting the data. The computation procedure is presented in (1) [23].

$$Latency = \frac{\mathrm{L}}{\mathrm{B}} \qquad (1)$$

Where: L: is packet length size (bits), and B: is used bandwidth (bits/s)

2.2.2. Average Throughput

Average Throughput is an indicator of the data processing ability of the suggested communication system. It reflects the amount of data that can be transferred or processed during a certain time interval, thereby reflecting the effectiveness of data transfer. This parameter allows analyzing the behavior of the system in case of alteration of the volume or number of transmitted messages. The larger the throughput, the larger the volume of the data transmitted during the certain time interval. In the suggested system, average data rate is estimated in bytes of second also depends on the ratio of the volume of the transferred data and the time needed for their transfer. It is calculated according to (2).

$$Average\ Throughput = \frac{\mathrm{D}}{\mathrm{T}} \qquad (2)$$

Where; D: is quantity of data transferred in bytes, and B: is Time taken for transfer in seconds

2.2.3 Packet Loss Ratio

Packet Loss Ratio (PLR) is adopted to estimate the percentage of packets that are not able to be delivered to their destinations. This metric shows how effective and reliable the connection is under different transmission conditions. The lower PLR means the lower percentage of lost data and, hence, the higher efficiency of delivery. In our research work, PLR is presented in percent form, and it is calculated as the ratio between transmitted and received packets. It is shown in (3) [24], [25].

$$PLR_{\%} = \left(1 - \frac{P_R}{P_S}\right) X\ 100 \qquad (3)$$

Where:P_R number of packets sent, and P_S number of packets received.

2.2.4 Packet Delivery Ratio

The Packet Delivery Ratio (PDR) refers to the ratio of successfully delivered packets to the total transmitted packets. These metric acts as an important measure of assessing the efficiency of information transfer between the ESP32 sender and receiver. The higher the value of PDR, the higher will be the number of generated messages reaching the target node. In the suggested model, PDR is calculated in terms of the percentage of the packets successfully received to the total transmitted packets, as shown in equation (4) [25], [26].

$$PDR_{\%} = \left(\frac{P_R}{P_S}\right) X\ 100 \qquad (4)$$

Where:P_R number of packages received successfully, and P_S total number of packets sent.

2.2.5 Communication Availability

Availability is an attribute that shows the level of functioning of the communication system and its ability to perform the desired function in the considered observation period. Availability in the IoT monitoring framework is characterized by the share of information that successfully passes through the communication system. The result will be given in percentage form and will be calculated based on the ratio of total packets transmitted and the total packets (both transmitted and lost). The formula for availability is given in (5) [25], [27].

$$Availability_{\%} = \left(\frac{P_S}{2P_S - P_R}\right) X\ 100 \qquad (5)$$

Where: P_R Number of packages received, and P_S total number of packets sent.

2.2.6 Sensor Measurement Accuracy

The accuracies of the sensors are studied to establish how far away the volume measurements of water taken using an ultrasonic sensor differ from the volume measurements taken using the reference sensor. In the current research, the Roots-Mean-Squares-Error (RMSE) approaches are applied in evaluating the extent of the deviation of water level measurements from those taken using the reference sensor. [28]. Lower RMSE indicates lower deviation of the sensor readings from the actual value, hence high accuracy of the sensors. The RMSE calculation is done using equation (6) [28].

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{n} e_i^2}{n}} \qquad (6)$$

Where e_i is the difference between actual and measured value, and n is the number of observations/ measurements.

2. EXPERIMENT RESULTS AND ANALYSIS

This part gives the experimental results of the ESP32-based IoT water monitoring system and discusses its performance under different data transmission scenarios. The analysis will focus on the performance parameters explained above, thus allowing the effect of the transmission frequency of messages on the communication process to be studied. Four experiments will be carried out in order to measure the average delay, average throughput, average packet-loss-ratio (PLR), availability, and average packet-delivery-ratio (PDR) of the system and its reliability behavior. MQTT will be used as the communication protocol for the four transmission frequencies of 1, 2, 5, and 10 Hz.

As far as the experiment is concerned, Node-RED works as the subscriber in this case and this device is connected through a wired LAN connection to the network from the personal computer. The MQTT broker as well as the publisher created using NodeMCU with ESP32 are working in this case on the same laptop and communicating through the Wi-Fi connection. The MQTT broker works as the middleware between the publisher and the subscriber and the QoS value of this device is set to 0 in this case. Here the transmission rate refers to the number of MQTT messages that are sent per second.

Duration for each experiment is set to 60 seconds and the whole process will be carried out 30 times in order to gather measurement data needed for performance evaluation. Wireshark software tool will be used for capturing network traffic that occurs during the experiments. Measurement data gathered is then used to analyze behavior of the proposed system at different data rates. Results obtained from each experiment together with their analysis are provided In the subsequent sections:.

3.1 Approach 1: Evaluation MQTT Communication Latency

First test aims at measuring the communication delay that results when MQTT messages are transferred via the architecture of IoT. Three unidirectional communication channels are measured; these include the communication channel between the NodeMCU (ESP32) and the MQTT broker, the broker to Node-RED communication channel, and the direct NodeMCU to Node-RED communication channel. The delays experienced by these communication channels depending on different transmission rates are shown in Figure 4. Based on the experimental findings, communication latency changes according to the chosen transmission frequency. With the increase in the rate of the message from 1Hz to 10Hz, it has been seen that there is an increasing trend in the recorded latency. Though the transmission frequency increases the chances of generating and publishing messages, it also causes an increase in the communication latency. These findings are in line with those presented in [23] regarding the impact of increased data transmission rate on latency. The comparison of the system proposed in this study with the one tested in [23] shows that our prototype has a reduced latency while operating at 1, 2, and 5 Hz. From these findings, it can be concluded that the speed of message generation has a direct impact on the latency of the communication process in the system under consideration. Therefore, choosing the correct frequency of transmission is very important for achieving the required compromise in terms of communication performance.

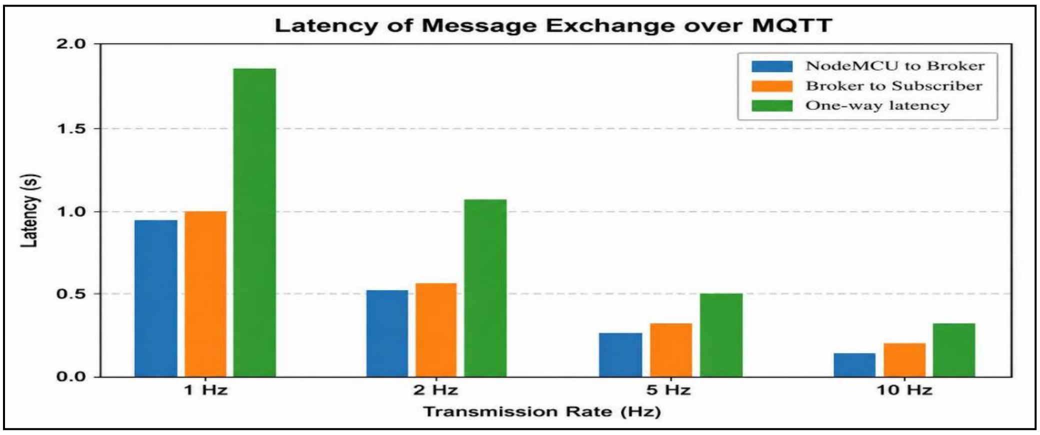

Figure 4. Latency of MQTT Communication with Different Message Transmission Rates

2.2 Approach 2: Evaluation MQTT Data transfer Throughput

This paper explores the connection between the rate of message transmission and the corresponding throughput achieved. Figure 5 below illustrate that the throughput of the proposed system is achieved at a transmission rate of 10Hz for both communication channels. From these experiments, it is clear that increasing the message transmission rate will increase the amount of data sent within the same period of time.

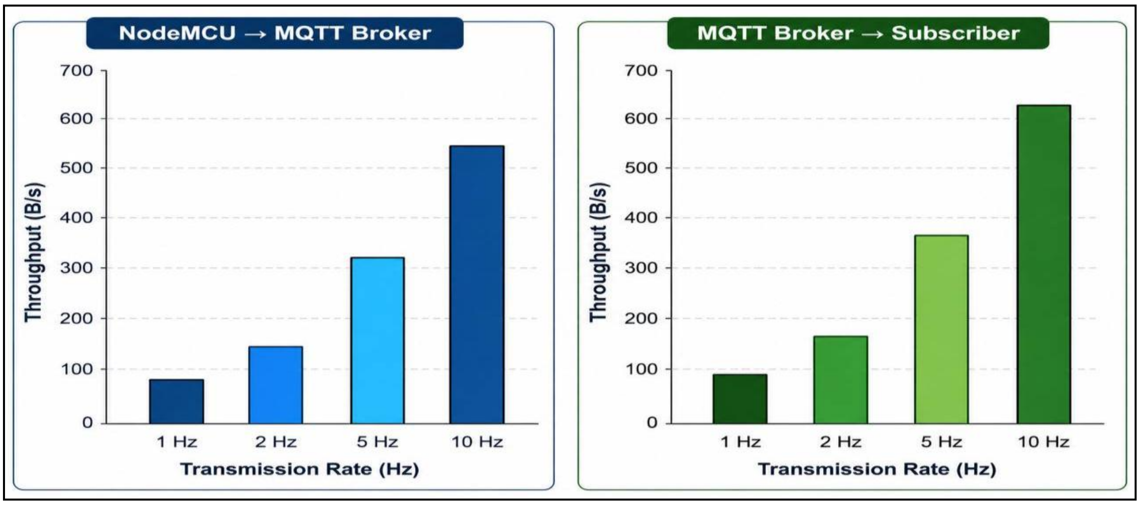

Figure 5. Throughput Performance of MQTT Communication Under Different Transmission

Rates

The transmission frequency refers to the number of data messages formed in one second, and hence an increase in the transmission frequency will result in a higher amount of data transmitted in the same observation period. Throughput is defined using Eq. (2) and is proportional to the quantity of information transmitted and the duration to transmit the same. In this current experiment, the time period of the transmission process for all rates is kept constant at 60 seconds. This means that any changes in throughput are caused mainly by the change in the data transmission rate since a higher transmission frequency results in more data being generated and transmitted in the same time.

2.3 Approach 3: Evaluation of Packet Delivery and Communication Availability

The third experiment studies the reliability of message delivery from the publisher side to the subscriber side based on different transmission rates. This experiment evaluates the following three parameters: availability, packet delivery ratio (PDR), and packet loss ratio (PLR). The results of the experiments performed at 1, 2, 5, and 10 Hz are listed in Table 1. The analysis of the results reveals that there is a direct correlation between the rate of message creation and the effectiveness of packet delivery. Thus, raising the value from 1 Hz to 10 Hz, one gets PLR growth, while both PDR and availability decrease. This tendency may be explained by the increased amount of packets transmitted per second, thus, raising the probability of losing packets during communication [23]. Additionally, the application of MQTT protocol with QoS level equals to zero does not guarantee delivery of packets to the receiver; therefore, it may result in failure of message delivery [29]. As a result, the combination of increased frequency of packet generation and particular QoS value leads to reduced PDR and availability. At the lowest rate of 1 Hz, the system achieves availability 97.8% and PDR 97.7%, which means PLR 2.2%.

Table 1. Communication Reliability Metrics at Different Transmission Reliability

Transmission Frequency (Hz)	Packet Delivery Ratio (PDR(	Packet Loss Ratio (PLR)	Communication Availability
1	97.7%	2.2%	97.8%
2	94%	5.9%	94.4%
5	84.6%	15.3%	86.6%
10	72.4%	27.5%	78.3%

2.4 Approach 4: Evaluation of Ultrasonic Sensor Measurement Accuracy

The fourth experiment is aimed at estimating the measurement performance of the ultrasonic sensor under different frequencies of message transmissions. The results of measuring performance are presented in Figure 7 at several different frequencies. As is clear from the results, there is a tendency that RMSE grows with an increase in the transmission frequency. Since RMSE reflects the distance between the obtained values and the reference ones, the lower RMSE means the better approximation to the real values of water levels and, consequently, the higher accuracy of measurements [30], [31]. The best results are obtained with the use of 1 Hz frequency, where RMSE is equal to 0.71, and accuracy is 82.77%. Therefore, the increasing tendency of RMSE with growth of the transmission frequency shows that message transmission frequency significantly affects the performance of ultrasonic sensors in measurements.

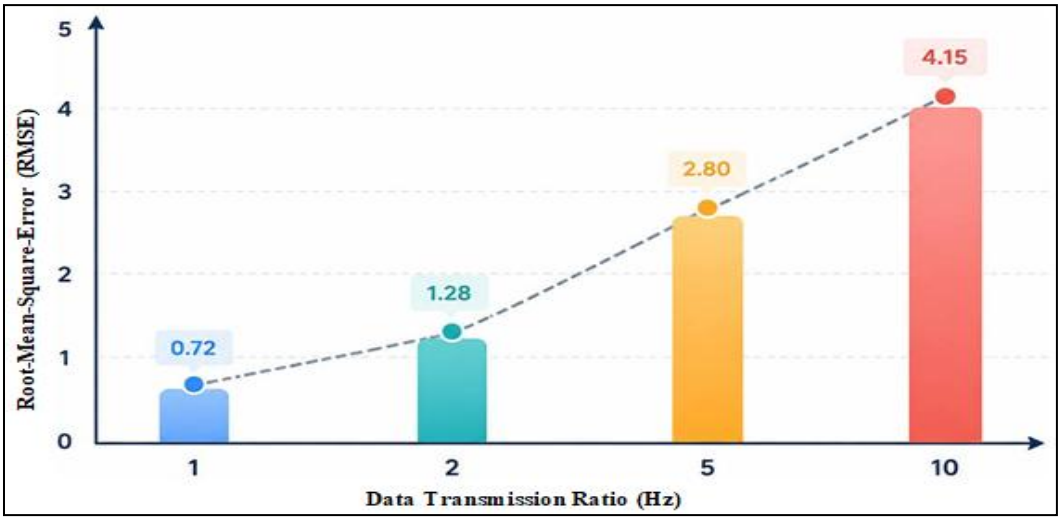

Figure 6. Impact of Data Transmission Ratio on RMSE

3. CONCLUSION

Thus, this study presents an IoT-based prototype for water level monitoring using ESP32 platform designed to explore the effect of varying the message transmission frequency on the reliability of communication via MQTT protocol. The experiment took into account several communication and sensing parameters such as latency, throughput, PLR, PDR, availability, and accuracy of measurement. As a result, the obtained findings prove the fact that the selected message transmission frequency has a considerable impact on the behavior of the considered system. At the lower transmission rate, the system ensures a high communication reliability, achieving approximately 98% in availability and PDR and maintaining a relatively low level of packet losses.

Moreover, the experiment shows that the change in the message transmission frequency leads to different outcomes in terms of all considered performance parameters. An increase in the transmission rate improves the throughput of the communication system while the other reliability-related measures, namely availability, PDR, and sensor accuracy, decline. In addition, it was found out that the selected transmission rate influences the latency of the process of communication through MQTT protocol. Consequently, it is necessary to select the proper transmission rate in order to meet the requirements of IoT monitoring applications in terms of reliability and timeliness of data delivery.

In turn, as a possible way for extending the considered system, it is necessary to implement several additional sensors and test the performance of the system in large-scale environments. Moreover, further research should take into account various environmental conditions and data traffic loads.

Future Work:

Future work would focus on improving the proposed architecture of water-level monitoring using IoT by intelligent and adaptive methods. For example, AI-based learning algorithms can be used applied to process the water-level measurements, detect anomalies in them and predict potential dangerous changes in the water level. Also, an adaptive communication scheme may be developed to adjust the rate of MQTT message transmission depending on detected water-level changes, the state of the network and application needs. This way it would be possible to reduce superfluous communication and still have timely monitoring data delivery. Moreover, more work can be done on developing light-edge intelligence solutions for the ESP32 class devices that should be taken into consideration due to their limited computational and energy capabilities. Usage of different types of heterogeneous sensors can give additional information about the environment and help in the detection of events. One possible idea is development of a flood warning predictive system that would use current sensor data, historical data and machine learning for prediction. Finally, the proposed solution can be tested with many nodes and different conditions.

REFERENCES

[1] L. González, A. Gonzales, S. Gonzalez Martinez, and A. Cartuche, "A Low-Cost IoT Architecture Based on LPWAN and MQTT for Monitoring Water Resources in Andean Wetlands," SN Computer Science, vol. 5, 2024, doi: 10.1007/s42979-023-02515-4.

[2] Z. Kegenbekov and A. Saparova, "Using the MQTT protocol to transmit vehicle telemetry data," *Transportation Research Procedia*, vol. 61, pp. 410–417, 2022, doi: 10.1016/j.trpro.2022.01.067.

[3] C.-R. Dow, D.-B. Nguyen, S. Cheng, P.-Y. Lai, and S.-F. Hwang, "VIPER: an adaptive guidance and notification service system in internet of vehicles," *World Wide Web*, vol. 22, no. 4, pp. 1669–1697, Jul. 2019, doi: 10.1007/s11280-018-0617-7.

[4] D. Silva, L. I. Carvalho, J. Soares, and R. C. Sofia, "A performance analysis of internet of things networking protocols: evaluating MQTT, CoAP, OPC UA," *Applied Sciences*, vol. 11, no. 11, May 2021, doi: 10.3390/app11114879.

[5] B. Mishra and A. Kertesz, "The use of MQTT in M2M and IoT systems: a survey," *IEEE Access*, vol. 8, pp. 201071–201086, 2020, doi: 10.1109/ACCESS.2020.3035849.

[6] B. Mishra, B. Mishra, and A. Kertesz, "Stress-testing MQTT brokers: a comparative analysis of performance measurements," *Energies*, vol. 14, no. 18, Sep. 2021, doi: 10.3390/en14185817.

[7] E. Bertrand-Martinez, P. Dias Feio, V. de Brito Nascimento, F. Kon, and A. Abelém, "Classification and evaluation of IoT brokers: a methodology," *International Journal of Network Management*, vol. 31, no. 3, May 2021, doi: 10.1002/nem.2115.

[8] F. Azzedin and T. Alhazmi, "Secure data distribution architecture in IoT using MQTT," *Applied Sciences*, vol. 13, no. 4, Feb. 2023, doi: 10.3390/app13042515.

[9] G. Pettorru and M. Martalo, "A Persistent and Secure Publish-Subscriber Architecture for Low-Latency IoT Communications," IEEE Transactions on Network and Service Management, vol. 23, pp. 729–739, 2025, doi: 10.1109/TNSM.2025.3635212.

[10] S. Quincozes, T. Emilio, and J. Kazienko, "MQTT protocol: fundamentals, tools and future directions," *IEEE Latin America Transactions*, vol. 17, no. 09, pp. 1439–1448, Sep. 2019, doi: 10.1109/TLA.2019.8931137.

[11] C. Wan, A. Mehmood, M. Carsten, G. Epiphaniou, and J. Lloret, "A blockchain based

forensic system for IoT sensors using MQTT protocol," in *2022 9th International Conference on Internet of Things: Systems, Management and Security (IOTSMS)*, Nov. 2022, pp. 1–8, doi: 10.1109/IOTSMS58070.2022.10062190.

[12] C. S. Yeh, S. L. Chen, and I. C. Li, "Implementation of MQTT protocol based network architecture for smart factory," in *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, Nov. 2021, vol. 235, no. 13, pp. 2132–2142, doi: 10.1177/09544054211014488.

[13] S. Jeddou, F. Fernández, L. Diez, A. Baina, N. Abdallah, and R. Agüero, "Delay and energy consumption of MQTT over QUIC: an empirical characterization using commercial-off-the-shelf devices," *Sensors*, vol. 22, no. 10, May 2022, doi: 10.3390/s22103694.

[14] A. Gavrilov, M. Bergaliyev, S. Tinyakov, K. Krinkin, and P. Popov, "Using IoT protocols in real-time systems: protocol analysis and evaluation of data transmission characteristics," *Journal of Computer Networks and Communications*, vol. 2022, pp. 1–18, Aug. 2022, doi: 10.1155/2022/7368691.

[15] K. T. M. Tran, A. X. Pham, N. P. Nguyen, and P. T. Dang, "Analysis and Performance comparison of IoT message transfer protocols applying in real photovoltaic system," *International Journal of Networked and Distributed Computing*, vol. 12, no. 1, pp. 131–143, Jun. 2024, doi: 10.1007/s44227-024-00021-4.

[16] H. Y. Chien *et al.*, "A MQTT-API-compatible IoT security-enhanced platform," *International Journal of Sensor Networks*, vol. 32, no. 1, 2020, doi: 10.1504/IJSNET.2020.104463.

[17] S. L. Mohammed, A. Al-Naji, M. M. Farjo, and J. Chahl, "Highly accurate water level measurement system using a microcon-troller and an ultrasonic sensor," *IOP Conference Series: Materials Science and Engineering*, vol. 518, no. 4, May 2019, doi: 10.1088/1757-899X/518/4/042025.

[18] A. Anggrawan, S. Hadi, and C. Satria, "IoT-based garbage container system using NodeMCU ESP32 microcontroller," *Journal of Advances in Information Technology*, vol. 13, no. 6, 2022, doi: 10.12720/jait.13.6.569-577.

[19] N. Kurniawan, "Electrical energy monitoring system and automatic transfer switch (ATS) controller with the internet of things for solar power plants," *Journal of Soft Computing Exploration*, vol. 1, no. 1, Sep. 2020, doi: 10.52465/joscex.v1i1.2.

[20] R. Lakshmanan, M. Djama, S. Perumal, and R. Abdulla, "Automated smart hydroponics system using internet of things," *International Journal of Electrical and Computer Engineering*, vol. 10, no. 6, pp. 6389–6398, Dec. 2020, doi: 10.11591/ijece.v10i6.pp6389-6398.

[21] D. Torres, J. P. Dias, A. Restivo, and H. S. Ferreira, "Real-time feedback in Node-RED for IoT development: an empirical study," in *2020 IEEE/ACM 24th International Symposium on Distributed Simulation and Real Time Applications (DS-RT)*, Sep. 2020, pp. 1–8, doi: 10.1109/DS-RT50469.2020.9213544.

[22] A. D. Salman, "IoT monitoring system based on MQTT publisher/subscriber protocol," *Iraqi Journal of Computer, Communication, Control and System Engineering*, pp. 75–83, Jul. 2020, doi: 10.33103/uot.ijccce.20.3.7.

[23] A. A. Zulfikri, D. Perdana, and G. Bisono, "Design and analysis of trash monitoring system prototype based on internet of things (IoT) using MQTT protocol," *Jurnal Infotel*, vol. 10, no. 3, Aug. 2018, doi: 10.20895/infotel.v10i3.381.

[24] A. Mamane, M. Fattah, M. El Ghazi, M. El Bekkali, Y. Balboul, and S. Mazer, "Scheduling algorithms for 5G networks and beyond: classification and survey," *IEEE Access*, vol. 10, pp. 51643–51661, 2022, doi: 10.1109/ACCESS.2022.3174579.

[25] B. Mbarek, M. Ge, and T. Pitner, "An efficient mutual authentication scheme for

internet of things," *Internet of Things*, vol. 9, Mar. 2020, doi: 10.1016/j.iot.2020.100160.

[26] R. Ratnasih, D. Perdana, and Y. G. Bisono, "Performance analysis and automatic prototype aquaponic of system design based on internet of things (IoT) using MQTT protocol," *Jurnal Infotel*, vol. 10, no. 3, 2018, doi: 10.20895/infotel.v10i3.388.

[27] M. F. Khan, E. A. Felemban, S. Qaisar, and S. Ali, "Performance analysis on packet delivery ratio and end-to-end delay of different network topologies in wireless sensor networks (WSNs)," in *2013 IEEE 9th International Conference on Mobile Ad-hoc and Sensor Networks*, Dec. 2013, pp. 324–329, doi: 10.1109/MSN.2013.74.

[28] A. R. Abdellah and A. Koucheryavy, "Deep learning with long short-term memory for IoT traffic prediction," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 12525 LNCS, pp. 267–280, 2020, doi: 10.1007/978-3-030-65726-0_24.

[29] H. H. Alshammari, "The internet of things healthcare monitoring system based on MQTT protocol," *Alexandria Engineering Journal*, vol. 69, pp. 275–287, Apr. 2023, doi: 10.1016/j.aej.2023.01.065.

[30] A. Barthwal and D. Acharya, "An IoT based sensing system for modeling and forecasting urban air quality," *Wireless Personal Communications*, vol. 116, no. 4, pp. 3503–3526, Feb. 2021, doi: 10.1007/s11277-020-07862-6.

[31] M. M. Jasim and A. M. Saed, "A study of the performance WSN in hospitals for patient monitoring," AIP Conference Proceedings, vol. 3264, 020017, 2025, doi: 10.1063/5.0259422.