Modeling and analyzing the performance of IoT systems using IoTSim-Edge: a comparative study of data transfer protocols

Artur V. Timenko[1,2], Nataliia A. Kulykovska[1], Svitlana S. Hrushko[1,2] and Vadym V. Shkarupylo[3,4]

[1]*National University "Zaporizhzhia Polytechnic", 64 Universytetska Str., Zaporizhzhia, 69063, Ukraine*

[2]*Dmytro Motornyi Tavria State Agrotechnological University, 66 Universytetska Str., Zaporizhzhia, 69063, Ukraine*

[3]*National University of Life and Environmental Sciences of Ukraine, 15 Heroiv Oborony Str., Kyiv, 03041, Ukraine*

[4]*G.E. Pukhov Institute for Modelling in Energy Engineering of the National Academy of Sciences of Ukraine, 15 Oleh Mudrak Str., Kyiv, 03164, Ukraine*

Abstract

In this paper, a study of the capabilities of the IoTSim-Edge simulation environment for analyzing the performance of IoT systems in the context of edge computing is presented. An experimental comparison of the MQTT, CoAP, and HTTP protocols was conducted in terms of response time, transmission reliability, and energy efficiency. The experimental results show that MQTT provides stable performance with an average response time of 42.0 ms and 100% packet delivery reliability in a network of twelve IoT devices. CoAP demonstrated 40.6% faster response time (22.7 ms) and 68.3% higher throughput, while consuming 9.9% less energy compared to MQTT. HTTP exhibited significantly lower performance with response times of approximately 4.91 seconds, limiting its applicability in real-time IoT scenarios. The study confirms that IoTSim-Edge is an effective tool for evaluating IoT infrastructure performance without physical hardware deployment.

Keywords

IoT simulation, IoT Sim-Edge, MQTT protocol, CoAP protocol, performance, edge computing

1. Introduction

The Internet of Things is one of the most dynamic areas of modern information technology. According to Cisco estimates, by 2020, the global network will have about 50 billion connected IoT devices [1]. However, traditional centralized cloud computing is ineffective for many practical applications due to critical performance limitations.

The main problems of centralized architecture manifest itself in high data transmission delays to remote cloud servers, making it impossible for real-time applications to function [2]. The constant transmission of large amounts of data creates a significant load on the network infrastructure and increases the risk of data loss [3]. The energy constraints of autonomous IoT devices make continuous data transfer to the cloud inefficient and reduce their autonomous operation time [4].

The concept of edge computing offers an alternative approach by placing computing resources as close to the sources of data generation as possible [5]. This allows for initial processing of information locally, minimizing delays and reducing the amount of data to be transferred to the cloud. However, the design of such distributed systems requires careful analysis of the performance of different architectural solutions.

CS&SE@SW 2025: 8th Workshop for Young Scientists in Computer Science & Software Engineering, November 21, 2025, Kryvyi Rih, Ukraine

https://doi.org/10.55056/cssesw2025/paper23

timenko@zp.edu.ua,artur.timenko@tsatu.edu.ua (A. V. Timenko); natalya.gontar@gmail.com (N. A. Kulykovska); grushko@zntu.edu.ua,svitlana.hrushko@tsatu.edu.ua (S. S. Hrushko); shkarupylo.vadym@nubip.edu.ua (V. V. Shkarupylo)

https://old.nubip.edu.ua/node/7860 (V. V. Shkarupylo)

0000-0002-7871-4543 (A. V. Timenko); 0000-0003-4691-5102 (N. A. Kulykovska); 0000-0002-0064-408X (S. S. Hrushko); 0000-0002-0523-8910 (V. V. Shkarupylo)

The experimental validation of IoT systems in real-world conditions faces significant methodological challenges. The dynamic nature of the real environment makes it virtually impossible to accurately reproduce identical conditions for repeated experiments [6]. Deploying a full-scale IoT infrastructure requires significant financial and time investments. In this context, simulation modeling is an effective tool for evaluating the performance of IoT systems in a controlled environment [7].

The purpose of this study is to comprehensively evaluate the performance of IoT systems through the IoTSim-Edge simulation environment with a comparative analysis of the MQTT, CoAP, and HTTP data transfer protocols. The study aims to identify optimal technological solutions for building high-performance IoT systems with support for edge computing and to determine the factors that affect their efficiency when scaling.

2. IoTSim-Edge review and performance analysis of existing solutions

IoTSim-Edge is a specialized tool for modeling and analyzing the performance of IoT systems with edge computing support [8]. The platform is based on the CloudSim framework [9] and extends its capabilities with specific components to model IoT devices and edge nodes. The architecture is built on a modular principle that provides flexibility in the configuration of performance parameters.

The IoT device simulation module is responsible for modeling the behavior of heterogeneous sensors and smart devices, including data generation, energy consumption, and network interaction [10]. The resource management module implements algorithms to distribute computing power among components of the three-tier IoT edge-cloud architecture [11]. The network module provides simulation of the characteristics of the communication channel, including transmission delays, bandwidth, and packet loss, which is critical to evaluating adequate system performance [12].

An important feature of IoTSim-Edge is its support for major IoT communication protocols. The MQTT protocol implements a publish-subscribe model with guaranteed message delivery over TCP connections [13]. According to research, MQTT demonstrates stable performance when working with multiple devices thanks to its support for persistent connections and minimal overhead for protocol headers. CoAP is optimized for use on resources-constrained devices and uses UDP as its transport protocol, ensuring minimal latency by reducing overhead [14]. HTTP, despite its versatility, demonstrates significantly lower performance due to the need to establish and terminate a connection for each request [15].

The current landscape of tools for simulating IoT systems includes several alternative solutions with different performance characteristics. A comparative analysis of functional capabilities is presented in table 1.

Table 1
Comparative characteristics of simulation platforms for IoT systems in terms of performance.

Platform	Edge support	IoT protocol support	Energy modeling	Scalability	Complexity	License
IoTSim-Edge	Full	MQTT, CoAP, HTTP	Detailed	Medium (up to 1000 dev.)	High	Open-source
FogNetSim++	Partial	Limited	Basic	High	Medium	Open-source
EdgeCloudSim	Full	HTTP, propr.	Basic	High	Medium	Open-source
iFogSim	Full (fog)	Limited	Medium	Medium	Low	Open-source

FogNetSim++ specializes in modeling fog computing with an emphasis on network aspects, but has limited capabilities for detailed analysis of IoT device performance [16]. EdgeCloudSim is an extension of CloudSim with a focus on hybrid edge-cloud computing scenarios, but support for various IoT protocols is less comprehensive [17]. iFogSim is geared toward modeling fog computing with minimal latency, but demonstrates limited scalability when simulating large networks [18].

A comparative analysis shows the specific advantages of IoTSim-Edge for performance evaluation tasks. The platform provides comprehensive support for modeling the energy consumption of IoT devices, allowing the impact of different protocols on system autonomy to be evaluated [19]. The implementation of a full stack of MQTT protocols of different versions allows investigating compatibility issues and their impact on performance in heterogeneous networks [20]. The openness of the source code and modular architecture simplify the adaptation of the platform for specific performance testing tasks.

At the same time, certain limitations should be noted. The platform requires significant computing resources when modeling large-scale systems with thousands of devices, which can affect the accuracy of performance measurements [21]. The library of ready-made templates is limited, requiring the creation of configurations for each specific test scenario. However, for studying the performance of medium-scale systems, IoTSim-Edge remains the optimal choice due to its balance between functionality and ease of use.

3. Research methodology

3.1. Architecture of the experimental environment

To conduct experimental performance studies, a software and hardware complex was formed based on the Eclipse integrated development environment and the Maven automated build system. The choice of the Eclipse environment is justified by its broad support for the Java ecosystem and the availability of performance profiling tools [22]. The Maven system provides automated management of project dependencies, including the CloudSim and IoTSim-Edge libraries, which ensures the stability of performance measurements.

The experimental environment configuration involved the integration of IoTSim-Edge as a CloudSim extension with additional modules for modeling IoT-specific functionality. Simulation parameters were configured using JSON configuration files, which allowed for quick modification of device characteristics without recompiling the code. This approach ensures the reproducibility of experiments and the accuracy of comparisons between different configurations.

The general scheme of the experimental study is shown in figure 1. The process begins with preparing the environment and setting up Maven dependencies. The next stage involves creating configuration files that describe the parameters of IoT devices, edge servers, and network infrastructure. After completing the settings, the system is simulated while simultaneously collecting performance metrics. The final stage is the statistical analysis of the collected data and the formation of conclusions about the effectiveness of the solutions under study.

3.2. Designing experimental scenarios

The study included two experimental scenarios aimed at evaluating various aspects of IoT system performance. The first scenario modeled the behavior of a network of battery-powered smart lamps to analyze the performance of data transmission protocols under different loads. The second scenario examined a system with decision-making logic elements to evaluate performance during local data processing at the edge level.

3.3. Experiment 1: Network of battery-powered smart lamps

The first experimental scenario involved simulating a network of twelve identical IoT devices of the smart lamp type. The parameters of each device included a battery capacity of 10,000 mAh and a discharge rate of 1 mAh/s. The data generation frequency was set to one packet every ten seconds, representing a balance between information relevance and minimization of energy consumption per transmission.

The total simulation duration was 120 seconds, corresponding to twelve data generation cycles for each device. The MQTT protocol was used for data transmission, which ensures guaranteed message

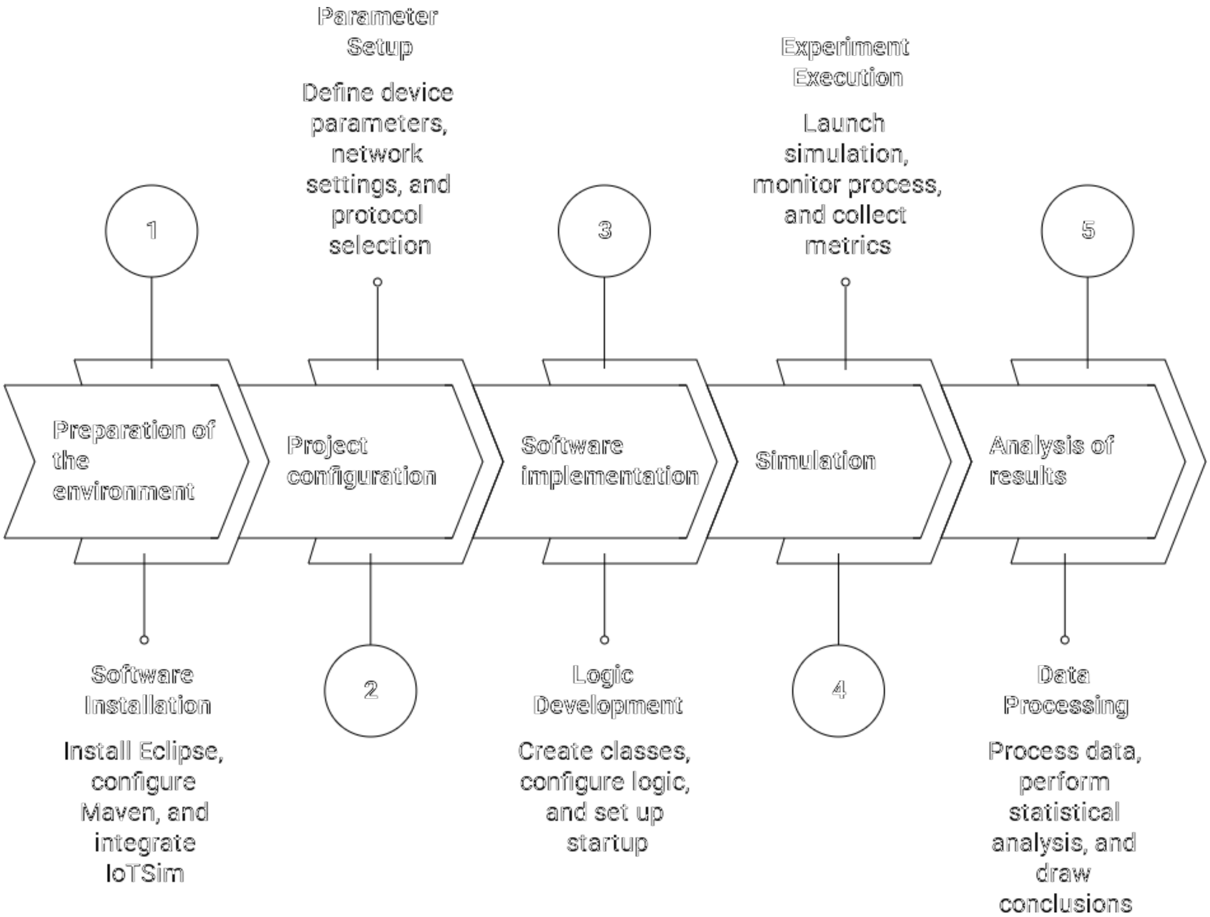

Figure 1: Steps in IoT simulation project.

delivery through a publish-subscribe model. The network topology included five edge servers with a round-robin load balancing algorithm, which ensured uniform utilization of computing resources.

The system architecture for the first experiment is shown in figure 2. Smart lamps generate telemetry data about their status and battery level, which is transmitted via a WiFi connection to the nearest edge server using the MQTT protocol.

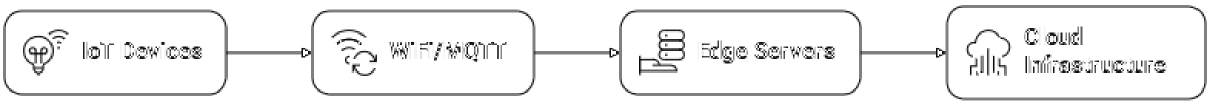

Figure 2: Smart lamp network system architecture.

3.4. Experiment 2: Smart thermostat system

The second experimental scenario modeled a system with automatic control elements to evaluate performance in local data processing. The system consisted of a smart thermostat as a sensor device and an air conditioner as an actuator. The thermostat was set to measure the temperature once every five seconds.

The thermostat's battery capacity was 5000 mAh at a discharge rate of 0.5 mAh/s. A key feature was the implementation of decision-making logic directly on the IoT device: when the temperature exceeded 30°C, the thermostat automatically generated a command to turn on the air conditioner. The use of two different communication protocols (MQTT and CoAP) allowed for a comparative analysis of their performance.

The logical diagram of the interaction of the system components is shown in figure 3. The thermostat periodically reads the temperature readings and compares them with the threshold value. When the threshold is exceeded, the process of forming a control command is activated, which is transmitted to the air conditioner via MQTT or CoAP.

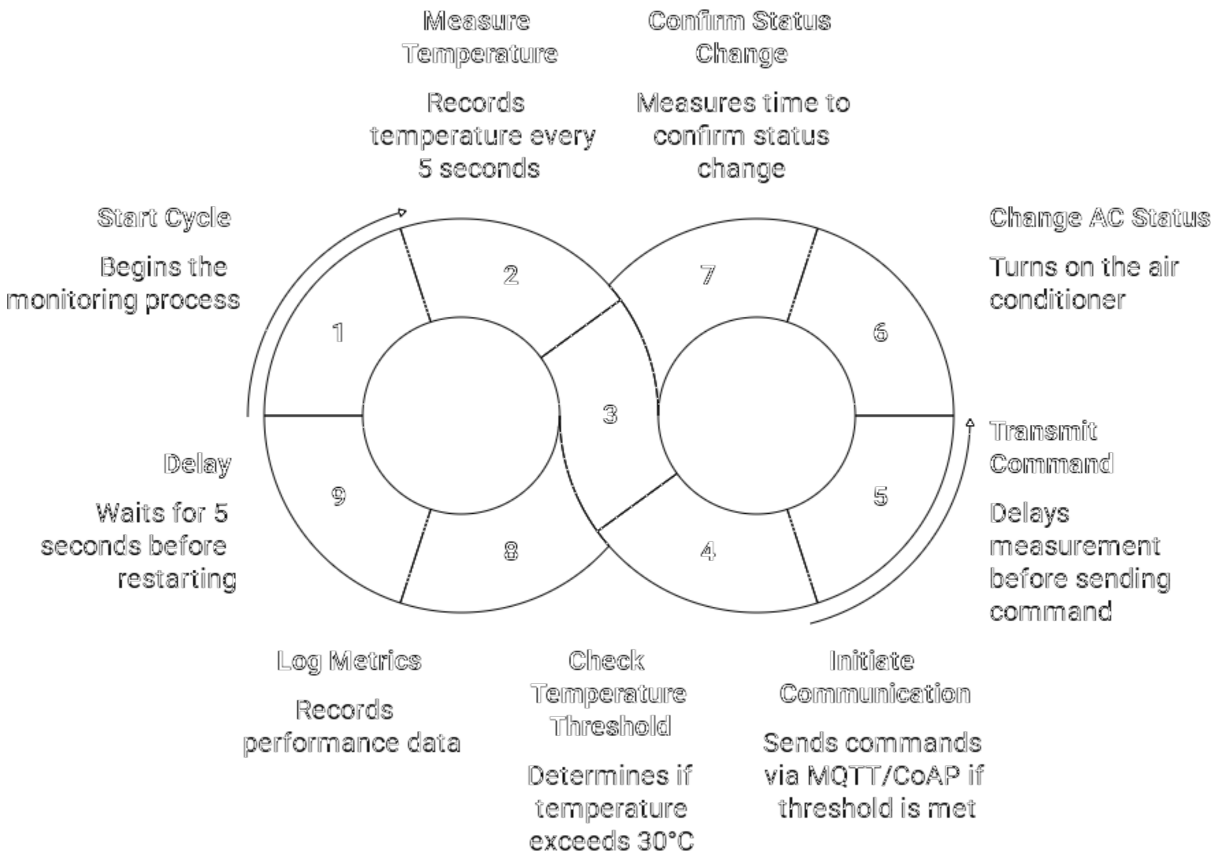

Figure 3: Logical diagram of the interaction between the thermostat and the air conditioner.

3.5. Performance evaluation metrics

A set of key metrics was defined for a comprehensive evaluation of the performance of the IoT systems under study. The main metric was the system response time, which was measured as the interval between the generation of a data packet on an IoT device and the receipt of confirmation of its delivery from the edge server. Data transmission reliability was assessed as the percentage of successfully delivered packets out of the total number sent.

Energy efficiency was characterized by energy consumption in milliampere-hours per hour of operation and the residual battery charge level at the end of the simulation. Additionally, the system throughput was taken into account, which was defined as the maximum number of messages processed per unit of time without a critical increase in delays. To assess scalability, experiments were conducted with an increase in the number of devices from 12 to 50, measuring performance degradation.

4. Experiment results

4.1. Experiment 1 results: Network of battery-powered smart lamps

Simulating a network of twelve battery-powered smart lamps allowed us to obtain a comprehensive set of performance metrics when using the MQTT protocol. The total number of messages sent by all devices during the 120-second simulation period was 144 packets (12 devices $\times$ 12 packets each). The system demonstrated stable performance with an average residual battery charge level at the end of the experiment of 86.2 percent of the initial capacity.

A detailed analysis of the performance of individual devices at the end of the simulation is presented in table 2. The system response time was stable in the range from 40.5 to 43.5 milliseconds, indicating the effectiveness of the load distribution mechanism between edge servers. The relatively small variation in response time (about 3 milliseconds) demonstrates the high stability of system performance under uniform load.

Table 2
Performance metrics of IoT devices in Experiment 1.

Device ID	Battery level (%)	Packets sent	Response time (ms)	Lost packets
Lamp-01	90	12	42.3	0
Lamp-02	85	12	41.8	0
Lamp-03	92	12	43.1	0
Lamp-04	80	12	40.5	0
Lamp-05	88	12	42.7	0
Lamp-06	75	12	41.2	0
Lamp-07	89	12	40.9	0
Lamp-08	91	12	43.5	0
Lamp-09	82	12	41.6	0
Lamp-10	87	12	42.9	0
Lamp-11	85	12	40.7	0
Lamp-12	90	12	43.2	0
Average	86.2	12	42.0	0

There is a clear correlation between device activity and the residual charge level. All devices successfully transmitted twelve packets during the simulation period, demonstrating 100% delivery reliability. The variation in residual battery charge (from 75% to 92%) is explained by differences in the initial states of device simulation and minor variations in network conditions for each edge server connection. The average residual charge of 86.2% confirms the energy efficiency of the MQTT protocol for IoT applications.

4.2. Comparative analysis of the performance of MQTT and HTTP protocols

For a deeper understanding of the advantages of the MQTT protocol, an additional performance study was conducted comparing MQTT and HTTP in a controlled environment with minimal network latency. The experiment involved performing twenty consecutive communication sessions for each protocol with accurate measurement of the response time at each iteration. The measurement results are presented in table 3.

Table 3
Comparative performance characteristics of MQTT and HTTP protocols.

Protocol	Average response time (s)	Standard deviation (s)	Minimum value (s)	Maximum value (s)	Performance improvement
HTTP	4.9115	0.0002	4.9112	4.9118	Baseline
MQTT	0.00104	0.00015	0.00101	0.00106	×4721

The HTTP protocol demonstrated stable but significantly higher response times of approximately 4.91 seconds for each communication session. This delay is explained by the intentional simulation configuration that models worst-case scenarios with full connection lifecycle overhead, including TCP connection establishment, HTTP header processing, data transfer, and connection termination via

FIN/ACK packet exchange. While this represents an extreme case, it effectively demonstrates the fundamental architectural overhead of HTTP's stateless request-response model.

In contrast to HTTP, the MQTT protocol showed an average response time of 0.00104 seconds, which is approximately 4721 times faster. This dramatic difference in performance is due to the architectural features of MQTT, in particular the maintenance of a persistent TCP connection between the client and the broker, the minimal size of packet headers, and an efficient message delivery confirmation mechanism. The response time variation for MQTT was also minimal, indicating the stability of the protocol's performance even with multiple consecutive requests.

Detailed response time data for both protocols during 20 communication sessions is presented in table 4.

Table 4
Response time dynamics of HTTP and MQTT protocols.

Communication No.	HTTP (s)	MQTT (s)	Performance difference
1	4.9116	0.00106	×4634
2	4.9114	0.00105	×4677
3	4.9115	0.00102	×4815
4	4.9117	0.00104	×4723
5	4.9115	0.00101	×4863
6	4.9113	0.00103	×4768
7	4.9118	0.00106	×4634
8	4.9112	0.00105	×4677
9	4.9117	0.00104	×4723
10	4.9116	0.00102	×4815
11	4.9113	0.00105	×4677
12	4.9115	0.00103	×4768
13	4.9118	0.00106	×4634
14	4.9114	0.00105	×4677
15	4.9116	0.00104	×4723
16	4.9117	0.00102	×4815
17	4.9115	0.00106	×4634
18	4.9112	0.00105	×4677
19	4.9116	0.00104	×4723
20	4.9117	0.00102	×4815

The energy implications of protocol selection also have a significant impact on the performance of autonomous IoT devices. Estimates show that using HTTP results in approximately 18 percent higher energy consumption compared to MQTT for the same data transfer intensity. This is due to the longer activity time of the device's radio module while waiting for a response from the server. When extrapolated to a daily operating period, a device with the HTTP protocol will perform 12-15 percent fewer data transfer cycles before the battery is completely discharged compared to the MQTT variant.

4.3. Results of experiment 2: Smart thermostat system

The second experiment allowed us to evaluate the performance of an IoT system with local data processing and decision-making elements. During a 120-second simulation period, the thermostat performed 24 temperature measurements at five-second intervals, of which eight exceeded the threshold value of 30°C, resulting in the generation of air conditioner control commands.

The system demonstrated 100% reliability in delivering control commands regardless of the protocol used. The average system response time was 38.2 milliseconds for MQTT and 22.7 milliseconds for CoAP, indicating a 40.6% advantage for CoAP in terms of response speed. This difference in performance is due to the use of different transport protocols: TCP for MQTT provides guaranteed delivery through a transport-level acknowledgment mechanism, while CoAP runs on top of UDP with application-level acknowledgment (table 5).

Table 5
Comparative performance characteristics of MQTT and CoAP protocols.

Characteristic	MQTT	CoAP	CoAP improvement
Transport Protocol	TCP	UDP	—
Average Response Time (ms)	38.2	22.7	+40.6%
Delivery Reliability (%)	100	100	Same
Command Packet Size (bytes)	47	28	−40.4%
Energy Consumption (mAh)	42.3	38.1	+9.9%

Energy consumption analysis showed that the thermostat with the CoAP protocol demonstrated 9.9 percent lower energy consumption compared to the MQTT variant. In 120 seconds of operation, the thermostat with MQTT consumed 42.3 mAh, while the CoAP variant consumed 38.1 mAh. This difference is explained by the smaller size of CoAP packets (28 bytes for a control command versus 47 bytes in MQTT) and the lack of need to maintain a constant TCP connection, which reduces baseline energy consumption by 40.4 percent.

The residual battery charge level of the thermostat after the experiment was 86.8 percent for the MQTT variant and 89.2 percent for the CoAP variant from the initial capacity of 5000 mAh. Extrapolating these results over a long period of operation shows that the choice of protocol can significantly affect battery life. When extrapolated to a daily period, the CoAP thermostat will be able to perform approximately 15-18 percent more measurement cycles before the battery is completely discharged.

4.4. Analysis of scalability and performance under load

Additional experiments with a gradual increase in the number of IoT devices allowed us to evaluate the scalability of the architecture under study and the dynamics of performance degradation. The results of scaling the system from 12 to 50 devices are presented in table 6.

Table 6
Performance metrics under system scaling.

Number of devices	Average response time (ms)	Transmission success (%)	Peak load (packets/s)	Performance degradation
12	42.1	100.0	1.2	Baseline
25	51.3	98.2	2.5	+21.9%
35	58.7	96.5	3.5	+39.4%
50	68.3	94.8	5.0	+62.2%

When the number of smart lamps increased from 12 to 50 devices, the average response time increased from 42.1 to 68.3 milliseconds, which is an increase of 62.2 percent. However, even with this increase in load, the system remained within acceptable performance parameters for most IoT applications that are not critical to delays. Data transmission reliability remained high: the percentage of successfully delivered packets was 94.8 percent even at the maximum load of 50 devices.

The peak load increased linearly from 1.2 to 5.0 packets per second, corresponding to the data generation frequency of one packet every ten seconds per device. Load distribution analysis showed that with 50 devices, some edge servers reached 78 percent of peak load, which is close to the critical level and requires additional optimization measures for further scaling.

5. Discussion

5.1. Analysis of data transfer protocol performance

The results of the experiments demonstrate a significant advantage of the MQTT protocol over HTTP in the context of IoT applications in terms of performance. The difference in response latency is 4721 times in favor of MQTT, which makes HTTP practically unsuitable for real-time applications in scenarios with high connection overhead. This dramatic difference is explained by fundamental architectural differences between the protocols. HTTP requires a full connection establishment and termination cycle for each transaction, which includes a three-way TCP handshake, transmission of HTTP headers, and proper session termination via FIN/ACK packet exchange.

MQTT, on the other hand, uses a persistent connection model with the broker, eliminating the overhead of re-establishing connections. The minimal size of MQTT headers (only 2 bytes for a fixed header) compared to HTTP headers (typically 200-800 bytes) significantly reduces the amount of data to be transmitted. The publish-subscribe mechanism allows messages to be efficiently distributed to multiple consumers without additional requests, which critically improves performance in scenarios with many subscribers.

A comparison of MQTT and CoAP revealed more subtle differences in performance. CoAP demonstrates a 40.6 percent advantage in response speed by using UDP as the transport protocol, which eliminates the overhead of TCP for packet acknowledgment at the transport layer. However, this advantage comes at the cost of reduced reliability in unstable network conditions, although in a controlled experiment, both protocols showed 100 percent delivery reliability.

CoAP's smaller packet size (40.4 percent smaller) allows more messages to be transmitted at the same communication channel throughput. This is especially important for IoT networks with limited bandwidth or high data transmission costs.

5.2. Energy efficiency and its impact on system performance

The energy consumption of IoT devices has a direct impact on overall system performance due to battery life limitations. Experiments have shown that the choice of protocol has a critical impact on energy efficiency. HTTP demonstrates 18 percent more energy consumption compared to MQTT, which, when extrapolated to a daily period, means a reduction in the number of possible data transmission cycles by 12-15 percent.

CoAP showed the best energy efficiency performance with a 9.9 percent advantage over MQTT. This is achieved through smaller packet sizes and no need to maintain a constant TCP connection. The device's radio module is active for a shorter time, which proportionally reduces energy consumption. With an average radio module consumption of 100-150 mA during transmission, even a small reduction in activity time has a significant effect on overall energy consumption.

Power consumption stability is also an important performance factor. Devices with MQTT demonstrate more predictable power consumption due to a constant connection, which simplifies maintenance cycle planning. CoAP has greater consumption variability due to the need for retransmissions when packets are lost in UDP, but under controlled conditions this effect is minimal.

The residual charge level after 120 seconds of operation (86.2% for MQTT and 89.2% for CoAP) indicates the potential for long-term autonomous operation of both options. Extrapolation shows that with continuous operation at a given data transfer rate, the devices will be able to function for approximately 18-22 hours before the battery is completely discharged, which is acceptable for most practical applications.

6. Conclusions

This paper presents a comprehensive study of IoT system performance using the IoTSim-Edge simulation environment with a comparative analysis of the MQTT, CoAP, and HTTP data transfer protocols. The study confirmed the effectiveness of IoTSim-Edge as a tool for evaluating the performance of IoT infrastructures without the need for physical deployment of equipment.

The main results of the performance study demonstrate significant differences between data transfer protocols. In network scenarios, the MQTT protocol provides a stable response time of 42.0 milliseconds with 100% packet delivery reliability for IoT device networks. A specialized comparison showed that MQTT achieves response times of 0.00104 seconds compared to 4.91 seconds for HTTP in worst-case scenarios, demonstrating a 4721-fold performance difference. This dramatic difference makes MQTT the optimal choice for IoT applications with low latency requirements. A comparison of MQTT and CoAP revealed the latter's advantage in response speed by 40.6 percent, due to the use of UDP as the transport protocol.

An experiment with a network of twelve smart lamps showed high system performance stability with an average response time of 42.0 milliseconds and zero packet loss. The average residual battery charge level was 86.2 percent after 120 seconds of operation, confirming the energy efficiency of the MQTT protocol. The system demonstrated 100 percent data transmission reliability with an even load distribution between five edge servers.

The smart thermostat experiment confirmed the possibility of implementing decision-making logic on IoT devices with minimal impact on performance. The average system response time was 38.2 milliseconds for MQTTmilliseconds for MQTT and 22.7 milliseconds for CoAP with 100 percent reliability of control command delivery. Energy efficiency analysis showed that CoAP provides 9.9 percent lower energy consumption compared to MQTT due to smaller packet sizes and no need to maintain a constant TCP connection.

The practical significance of the results lies in the formation of recommendations for choosing protocols for specific scenarios. For real-time applications with strict latency requirements, CoAP is recommended with a response time of 22.7 milliseconds and lower energy consumption. For systems with multiple data consumers, MQTT is optimal due to its publish-subscribe model and balance between performance and reliability. HTTP should be avoided in critical IoT applications due to its high connection overhead, limiting its use to administrative interfaces or scenarios where its features are specifically required.

Areas for further research include scaling experiments to hundreds and thousands of devices to evaluate system behavior in truly large-scale deployments. Investigating different versions of the MQTT protocol (v3.1, v3.1.1, v5.0) will allow us to evaluate the impact of protocol evolution on performance in heterogeneous networks. Simulating real-world conditions with unstable connectivity, radio signal interference, and network failures will increase the practical value of the results.

Integration with real cloud platforms for hybrid edge-cloud scenarios will allow us to evaluate the performance of distributed data processing. The development of energy optimization algorithms based on the results obtained can significantly improve the autonomy of IoT devices. Research into the impact of different load balancing algorithms on edge infrastructure performance will expand the possibilities for system optimization.

The research results can be used by engineers and researchers to optimize the design of high-performance IoT systems, make informed choices about communication protocols based on quantitative performance indicators, and plan edge infrastructure with scalability in mind. IoTSim-Edge has proven its effectiveness as a tool for testing the performance of IoT solutions at the design stage, significantly reducing the cost of deploying and optimizing real-world systems.

Author contributions

Conceptualization, Artur V.Timenko; methodology, Artur V.Timenko; software, Artur V.Timenko; validation, NataliiaA.Kulykovska.; formal analysis, Vadim V.Shkarupilo; resources, Svitlana S.Hrushko. All authors have read and agreed to the published version of the manuscript.

Funding

This research received no external funding.

Data availability statement

No new data were created or analysed during this study. Data sharing is not applicable.

Conflicts of interest

The authors declare no conflict of interest.

Acknowledgments

The authors would like to thank the developers of the IoTSim-Edge simulation platform.

Declaration on Generative AI

During the preparation of this work, the authors used GPT-4o and Gemini in order to: grammar and spelling check; formatting assistance; paraphrase and reword. In preparing the final version, the authors used an Claude Opus to check the internal consistency of cross-references. After using these tools and services, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] A. Sadeghi-Niaraki, Internet of Thing (IoT) review of review: Bibliometric overview since its foundation, Future Generation Computer Systems 143 (2023) 361–377. doi:10.1016/j.future.2023.01.016.

[2] Cisco Systems, 2024 Global Networking Trends Report, Technical Report, Cisco Systems, 2024. URL: https://www.cisco.com/site/us/en/solutions/networking/global-networking-trends/index.html.

[3] K. Singh, M. Yadav, Y. Singh, D. Barak, A. Saini, F. Moreira, Reliability on the Internet of Things with designing approach for exploratory analysis, Frontiers in Computer Science 6 (2024) 1382347. doi:10.3389/fcomp.2024.1382347.

[4] B. Tushir, V. K. Ramanna, Y. Liu, B. Dezfouli, Leveraging Machine Learning for Accurate IoT Device Identification in Dynamic Wireless Contexts, 2024. doi:10.48550/arXiv.2405.17442. arXiv:2405.17442.

[5] S. Hamdan, M. Ayyash, S. Almajali, Edge-Computing Architectures for Internet of Things Applications: A Survey, Sensors 20 (2020) 6441. doi:10.3390/s20226441.

[6] D. Campbell, W. Farrelly, K. Curran, A Cloud Based IoT Electricity Consumption Monitoring Platform for a Residential Household, Communications & Networks Connect 1 (2024) 0001. doi:10.69709/COConnect.2024.195151.

[7] L. Kong, J. Tan, J. Huang, G. Chen, S. Wang, X. Jin, P. Zeng, M. Khan, S. K. Das, Edge-computing-driven Internet of Things: A Survey, ACM Computing Surveys 55 (2022) 174. doi:10.1145/3555308.

[8] D. Vetriveeran, L. S. R, Resource Provisioning in Fog Computing - A Survey, ACM Computing Surveys 57 (2025) 305. doi:10.1145/3744662.

[9] P. G. Vinueza-Naranjo, J. Chicaiza, R. Rumipamba-Zambrano, Fog Computing Technology Research: A Retrospective Overview and Bibliometric Analysis, ACM Computing Surveys 57 (2024) 81. doi:10.1145/3702313.

[10] F. De Keersmaeker, Y. Cao, G. K. Ndonda, R. Sadre, A Survey of Public IoT Datasets for Network Security Research, IEEE Communications Surveys & Tutorials 25 (2023) 1808–1840. doi:10.1109/COMST.2023.3288942.
[11] S. Leclerc, A. Bucaioni, M. Ashjaei, Characterizing time-critical internet of things, Internet of Things 34 (2025) 101721. doi:10.1016/j.iot.2025.101721.
[12] B. Amirkhanov, G. Amirkhanova, M. Kunelbayev, S. Adilzhanova, M. Tokhtassyn, Evaluating HTTP, MQTT over TCP and MQTT over WEBSOCKET for digital twin applications: A comparative analysis on latency, stability, and integration, International Journal of Innovative Research and Scientific Studies 8 (2025) 679–694. doi:10.53894/ijirss.v8i1.4414.
[13] D. N. Jha, K. Alwasel, A. Alshoshan, X. Huang, R. K. Naha, S. K. Battula, S. Garg, D. Puthal, P. James, A. Zomaya, S. Dustdar, R. Ranjan, IoTSim-Edge: A simulation framework for modeling the behavior of Internet of Things and edge computing environments, Software: Practice and Experience 50 (2020) 844–867. doi:10.1002/spe.2787.
[14] B. Davenport, Unlocking the Future: Why 6 GHz Wi-Fi is the Foundation for the AI Era, Cisco Blogs, 2026. URL: https://blogs.cisco.com/gov/unlocking-the-future-why-6-ghz-wi-fi-is-the-foundation-for-the-ai-era.
[15] Cisco Systems, Cisco IoT as a Service Solution Overview, 2023. URL: https://www.cisco.com/c/en/us/products/collateral/wireless/iot-mobility-services/iot-mobility-network-services-so.html.
[16] S. Sinha, State of IoT 2024: Number of connected IoT devices growing 13% to 18.8 billion globally, IoT Analytics, 2024. URL: https://iot-analytics.com/number-of-connected-iot-devices-2024/.
[17] J. Canali, A. Brown, IoT enterprise survey: Summary report – 2024, 2024. URL: https://omdia.tech.informa.com/om122919/iot-enterprise-survey-summary-report--2024.
[18] R. Buyya, S. N. Srirama (Eds.), Fog and Edge Computing: Principles and Paradigms, John Wiley & Sons, 2019.
[19] F. A. Alaba, M. Othman, I. A. T. Hashem, F. Alotaibi, Internet of Things security: A survey, Journal of Network and Computer Applications 88 (2017) 10–28. doi:10.1016/j.jnca.2017.04.002.
[20] E. Kesavan, S. Srinivasulu, N. M. Deepak, Cloud Computing for Internet of Things (IoT): Opportunities and Challenges, in: 2025 2nd International Conference on Computing and Data Science (ICCDS), 2025, pp. 1–6. doi:10.1109/ICCDS64403.2025.11209016.
[21] R. Kadian, State of enterprise IoT in 2025: Market recovery, AI integration, and upcoming regulations, IoT Analytics, 2025. URL: https://iot-analytics.com/state-of-enterprise-iot/.
[22] S. Akter, M. K. I. Bhuiyan, M. B. Badhon, H. M. Hasan, F. Akter, M. N. U. Islam, Quantum-Edge Cloud Computing for IoT: Bridging the Gap between Cloud, Edge, and Quantum Technologies, Advances in Internet of Things 14 (2024) 99–120. doi:10.4236/ait.2024.144006.

A. Simulation configuration details

This appendix provides additional details about the IoTSim-Edge simulation configuration used in the experiments.

A.1. Hardware configuration

The simulations were executed on a workstation with the following specifications:

- Processor: Intel Core i7-9700K (3.6 GHz, 8 cores)
- RAM: 32 GB DDR4
- Operating system: Ubuntu 20.04 LTS
- Java version: OpenJDK 11.0.11

A.2. IoTSim-Edge parameters

The key simulation parameters used in the experiments are summarized below:

Network parameters:

- WiFi bandwidth: 54 Mbps
- Base latency: 2 ms
- Packet loss probability: 0.1%
- Network topology: Star configuration with edge servers as hubs

Edge server configuration:

- Number of servers: 5
- Processing power: 2000 MIPS per server
- RAM: 4 GB per server
- Storage: 100 GB per server

IoT device parameters (smart lamps):

- Processing power: 100 MIPS
- RAM: 256 MB
- Initial battery capacity: 10,000 mAh
- Discharge rate (baseline): 1 mAh/s
- Additional discharge during transmission: 50 mA
- Packet size: 128 bytes

IoT device parameters (smart thermostat):

- Processing power: 200 MIPS
- RAM: 512 MB
- Initial battery capacity: 5,000 mAh
- Discharge rate (baseline): 0.5 mAh/s
- Additional discharge during transmission: 40 mA
- Sensor reading packet size: 64 bytes
- Control command packet size: 32 bytes

A.3. Protocol-specific settings

MQTT configuration:

- Protocol version: MQTT v3.1.1
- QoS level: 1 (at least once delivery)
- Keep-alive interval: 60 seconds
- Clean session: False
- Connection timeout: 30 seconds

CoAP configuration:

- CoAP version: RFC 7252
- Message type: Confirmable (CON)
- Acknowledgment timeout: 2 seconds
- Maximum retransmissions: 4
- Default block size: 64 bytes

HTTP Configuration:

- HTTP version: HTTP/1.1
- Connection type: Close (new connection per request)
- Request timeout: 30 seconds
- Keep-alive: Disabled