Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

A Qualitative Comparative Analysis of MQTT, HTTP/HTTPS, CoAP, AMQP, and OPC UA for Internet of Things Applications

Clarizza O. Del Socorro[1], Alex Ajhaiezer D Susada[2], Mohamad D. Tapuli[3], Jay Ar P. Esparcia[4]
Department of Computer Engineering
University of Southern Mindanao
Kabacan, 9407, Philippines

Abstract **-** This study compares five commonly used application-layer communication protocols in Industrial Internet of Things (IIoT) systems: HTTP, MQTT, CoAP, AMQP, and OPC UA. As industrial environments continue to require faster, secure, scalable, and energy-efficient communication, choosing the right protocol is important to ensure good system performance and reliable operation. The paper discussed how each protocol is designed, how it communicates, how it sends data, and how it works. It also looks at key performance factors (Delay, Bandwidth use, Data speed, Resource use, Reliability, Security, and Scalability), when used or applied in industrial areas like smart manufacturing, predictive maintenance, remote monitoring, and system integration. Results indicate that MQTT has dynamic scalability, resource efficiency that provides quality level of service, and easy implementation, making it great for big systems and important tasks. CoAP is lightweight and very efficient in limited environments that can run on internet-connected devices where memory and computing resources are scarce. HTTP is widely compatible and easy to integrate with web systems, but it can use more bandwidth and may cause higher delays. AMQP provides reliable and organized messaging for enterprise systems with strong security, although it is more complex and uses more bandwidth. OPC UA is a very powerful server that can be used with full functionality, and ensure data integrity, trustworthiness and authorization within OPC communication relationships.

The study concludes by recommending protocol selection based on specific application requirements and suggests hybrid communication architectures as a future direction to optimize Industrial IoT system performance.

Keywords - MQTT, HTTP/HTTPS, CoAP, AMQP, and OPC UA.

I. INTRODUCTION

The rapid expansion of the Internet of Things and Industrial Internet of Things has increased the demand for reliable and efficient communication between devices, systems, and enterprise platforms. Modern industrial operations rely on the continuous exchange of data from sensors, controllers, cloud services, and management systems. The production lines, monitoring tools, and analytics platforms require accurate and timely information to maintain productivity and safety. When choosing an application layer protocol, latency, bandwidth usage, energy consumption, security level, and overall system performance are directly affected. Choosing the wrong protocol causes delays in communication, unstable connections, higher power consumption, and increased operational costs. The network conditions in IoT and IIoT environments are different. There are systems that use high-speed wired industrial networks, while others rely on wireless sensor nodes with limited battery life and low processing capacity. Due to these differences, there is no single communication protocol suitable for all situations.

The objective of this study is to identify research gaps of HTTP, MQTT, CoAP, AMQP, and OPC UA. This study aims to analyze their architectural designs and communication models to better understand their strengths, limitations, and application-specific suitability of each protocol in various IoT use cases including smart manufacturing, predictive maintenance, remote monitoring, and enterprise integration. It also compares performance characteristics such as latency, bandwidth consumption, and resource utilization in industrial environments. Finally, this study can provide recommendations and future research directions for optimizing protocol selection and hybrid communication architectures in Industrial IoT systems.

II. RELATED WORK

The performance and applicability of the commonly used IoT protocols such as MQTT, CoAP, and others.

Jain et al. [1] found that MQTT suits applications requiring reliable message delivery due to its QoS levels, while CoAP performs better for low-power and resource-constrained devices. Their results showed that MQTT processed larger payloads more effectively, whereas CoAP was more efficient in energy-limited setups.

Vashi et al. [2] studied transmission delay and energy usage. Their findings showed that CoAP reduces power consumption in low energy networks, whereas MQTT ensures steady and consistent data transmission. Our observations match their study. CoAP handled small data packets with lower energy demand, and MQTT maintained stable communication sessions.

Iglesias et al. [3] compared HTTP, CoAP, and MQTT in real time systems by measuring overhead and latency. They concluded that HTTP provides stronger security features, but CoAP and MQTT reduce communication overhead, which improves IoT performance. Our findings align with this outcome. HTTP3 delivered strong protection, while MQTT and CoAP showed better efficiency in time sensitive environments.

Ravi et al. [4] studied MQTT and CoAP in smart city systems. They reported that MQTT supports a larger number of

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

devices through its publish subscribe model, while CoAP performs better in direct and resource limited communication. Our results reflect this pattern. MQTT managed larger network setups more efficiently, and CoAP worked well in smaller IoT deployments.

Chen et al. [5] assessed several messaging protocols, including HTTP3. They found that HTTP3 offers stronger security and lower latency compared to HTTP 2, though it is heavier than CoAP. Our findings are consistent with this. HTTP3 provided a good balance between security and performance, while CoAP remained more suitable for constrained hardware.

Kumar et al. [6] examined MQTT and CoAP in industrial IoT, focusing on reliability and delay. Their study showed that MQTT delivers messages more reliably for critical industrial applications, whereas CoAP fits smaller and less complex networks. Our results support this conclusion. MQTT performed better in large and critical systems, while CoAP suited simple sensor networks.

Patel et al. [7] analyzed security aspects of MQTT and CoAP. They noted that MQTT uses TLS for strong protection, while CoAP relies on DTLS as a lighter security option for constrained environments. Our study also observed that CoAP's lightweight security features are effective in IoT systems with limited resources.

Rudevdagva and Riunaa [8] compared real-time communication protocols including OPC UA PubSub and AMQP. They noted that OPC UA excels in interoperability and industrial standard compliance, whereas AMQP provides higher scalability for large messaging systems

There has been a lot of research conducted that compares lightweight IoT protocols like MQTT and CoAP, but most studies only look at two or three protocols at a time and mostly look at settings that are limited or smart cities. Not much study has been done that gives a full comparison of both lightweight protocols and industrial-grade communication standards like OPC UA and AMQP in a single analytical framework. Also, there isn't a uniform evaluation of architectural design, scalability, security mechanisms, and how well they work for industrial deployment. This hole shows that we need a bigger comparison study that looks at HTTP, MQTT, CoAP, OPC UA, and AMQP in a more organized way. This will help people choose the right protocol for their different IoT systems.

III. METHODOLOGY

This study is qualitatively based comparative approaches with the existing research to analyze and evaluate IoT communication protocols based on existing research findings. Information gathered in this study was collected from different sources of information. The information gathering is conducted using key word "IoT communication protocols, analysis of HTTP and MQTT, what is HTTP, MQTT, CoAP, AMQT, and OPC UA."

The only research paper included in this study is between 2015 to 2026. Articles that are older than 10yrs, non academic sources, not written in English, and not directly related to IoT communication were excluded.

A total of 30 research papers were selected for the study. The protocols were compared based on their Bandwidth usage, Latency, Transport protocol, communication model, Reliability, security transport, scalability, and lastly Best application uses.

IV. COMPARATIVE ANALYSIS

HTTP (Hyper Text Transfer Protocol) - In 1989–1991 HTTP was developed by Tim Berners-Lee also the one who developed the World Wide Web and the first web browser [12]. HTTP works on a client-server approach and is the main way that web browsers and servers communicate to each other so that they can share information over the internet and it is also the standard protocol for transferring or exchanging over a web [9], [10]. HTTP is typically implemented over TLS (HTTPS), a more secured version of it [11].

Fig. 1. HTTP architecture overview [12]

Advantages

- It works on any device which means it is universally supported. [10]
- Data It's easy to understand and implement in the web application. [13]
- Has a HTTPS version which encrypt the content.
- Suitable for heavy weight uses. [9]

Disadvantages:

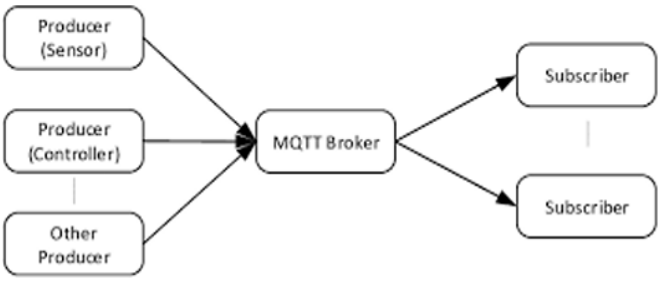
- Any request–response communication can cause delays. [14]
- Not designed or suited for continuous, low-latency communication. [15]
- HTTP has little or no memory of previous requests between the client and server. [9]
- Headers are provided or sent with each request and response, which increases bandwidth usage. [14]
- Without HTTPS it lacks security because data is transmitted and received in plain text and can be intercepted. [11]

MQTT - Message Queuing Telemetry Transport (MQTT) is a lightweight messaging protocol that uses a publish subscribe model, making it well-suited for reliable communication in networks that have limited bandwidth, high latency, or intermittent connectivity [9]. [14] It allows devices, sensors, and applications to send and receive data through a central broker, which separates the message publishers from the subscribers [14]. MQTT offers three Quality of Service

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

(QoS) levels—0 (at most once), 1 (at least once), and 2 (exactly once) [14]—so developers can choose the right balance between reliability and network efficiency.

Fig. 2. MQTT Architecture Overview. [16]



Advantages

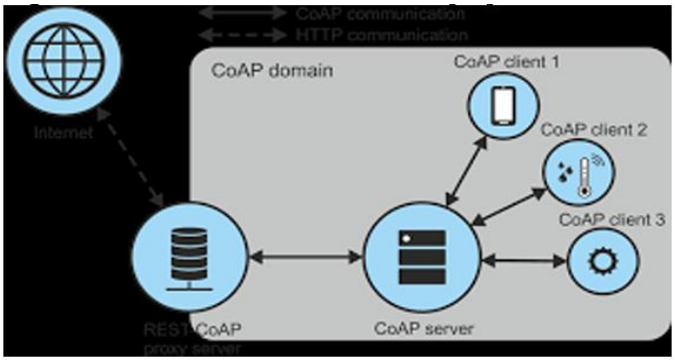
- Lightweight and efficient - uses small messages, which helps save bandwidth and makes communication faster. [17]
- Strong ecosystem support - MQTT is broadly absorbed or adopted across cloud platforms, and edge system, creating from integration into IoT surrounding connection straightforward. [9]
- Flexible publish–subscribe design - MQTT uses a broker to connect senders and receivers [14], making it easier to add more devices and manage communication between them.

Disadvantages

- High Communication Overhead - It uses TCP/IP and TCP requires more communication capabilities unlike UDP. [9]
- Scalability Limitation / Bottleneck - There is a limited capacity for communication for Broker. [18]
- Single Point of Failure (SPOF) - All nodes are connected to Broker. Therefore, communication collapses when the broker is a failure. [14]

CoAP - developed in 2014 to address the challenges of limited bandwidth, memory, and processing power, enabling IoT devices to communicate effectively in constrained environments [11], [10]. CoAP has greatly benefited IoT devices by offering numerous advantages that have sped up innovation, leading to the creation of new products and the growth of emerging industries. CoAP provides an efficient method for IoT devices to exchange data. it also uses UDP for transport protocol [11], and It allows IoT devices to exchange information seamlessly, supporting tasks like sending sensor data, receiving instructions, and coordinating within the broader IoT network.

Fig. 4. CoAP Architecture Overview. [19]



Advantages

- Minimal bandwidth use - CoAP uses a small message format that lowers network data traffic [11]. This helps IoT devices communicate even on slow or restricted connections.
- Optimized for constrained hardware [10] - Works well on devices with limited memory, processing power, and battery capacity. This makes it suitable for sensors and embedded systems.
- Quick data exchange [17] - The protocol enables fast transmission of sensor readings, commands, and control signals with very little delay.
- RESTful design - Built on REST principles similar to web communication [10],
- Energy-saving operation [20]- By minimizing processing and transmission demands, CoAP helps extend the battery life of wireless IoT devices.
- Scalable networking [9] - It supports communication among large numbers of devices, allowing IoT systems to expand without major redesigns.

Disadvantages

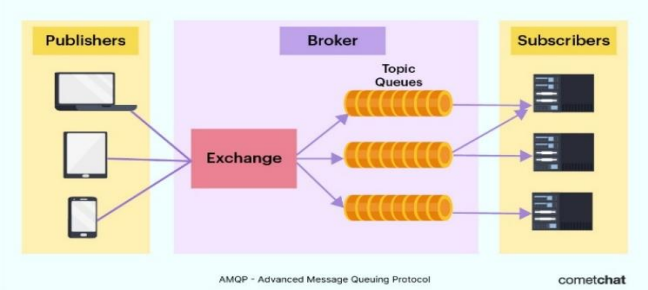
- Messages are unreliable [11] - Therefore, ACK (acknowledgement) packets are sent to confirm the message arrives. However, it does not show clearly whether these messages are decoded correctly or completely.
- It is still standardizing [9]- It is selected the most unstandardized protocol among other protocols.

AMQP (Advanced Message Queuing Protocol) - a TCP based messaging standard that supports multiple communication models, including publish subscribe like MQTT [9], [14]. It allows asynchronous message delivery, so systems running on different platforms, hardware, and programming languages can communicate smoothly. AMQP improves server efficiency through structured data framing and buffering methods. It also protects data using TLS for encryption and SASL for authentication.

An AMQP broker manages the communication between exchanges and queues [14]. The exchange evaluates these keys and directs the messages into the appropriate queues. Messages stay in the queues until subscribers retrieve them. Bindings set the connection rules between exchanges and queues. AMQP defines three exchange types, direct, fanout, and topic, which control message delivery.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

Fig. 3. AMQP architecture Overview. [21]



Advantages

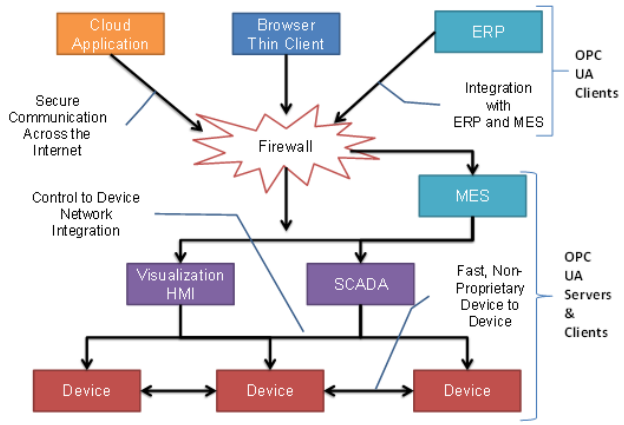
- Reliable message delivery- It uses QoS and hence ensures safe passage of important data [14].
- Supports publish–subscribe communication [9]- AMQP uses already established publish/subscribe architecture for data sharing as used by MQTT protocol.
- Strong interoperability [9] - It ensures interoperability as it uses wire level protocol which sends data as stream of bytes.
- Flexible communication options [14] - It offers simpler peer to peer communication along with intermediaries.
- Secure connections [11]- It offers secured connection to users using SSL protocol.

Disadvantages

- Not backward compatible- It is not backward compatible with old versions.
- More complex than simpler protocols [9] - It is not as simple as HTTP or any other wire protocols.
- Higher bandwidth usage [22]- It requires higher bandwidth.
- No built-in resource discovery [9]- Resource discovery is not supported.

OPC Unified Architecture (UA) - a communication technology standard introduced by the OPC Foundation in 2006 as an enhanced and more advanced version of the earlier OPC Classic standard. OPC is a standardized interface that enables communication among various data sources, such as factory floor devices [23], [24], laboratory instruments, test system equipment, and databases. This protocol relies on Microsoft's COM/DCOM technology to deliver standardized specifications for Data Access (DA), Historical Data Access (HDA), and Alarms and Events (A&E). OPC UA can be applied in supervisory control systems, removing the need for Windows-based intermediary systems and enabling a more efficient vertical flow of data from the field and control levels up to the management and enterprise levels.

Fig. 5. OPC UA Architecture Overview. [25]



Advantage

- Platform Independence [24] - Supports Windows, Linux, macOS, and embedded systems. You can connect different operating systems in one industrial setup without conflict.
- Robust Security - Applies encryption to protect data during transmission, verifies user identity through [11], [24]These measures prevent unauthorized access, reduce security risks, and keep communication between devices and systems protected.
- Compatibility - Connects devices, sensors, databases, and enterprise software without complex custom coding [26]. You reduce integration time and avoid constructing separate interfaces for each system.
- Expandability - Works for small embedded controllers and also for large enterprise level deployments [24]. You can start small and expand your system as your operations increase.
- Real time and Historical Data Access [24] - Gives you access to live process data and stored records. You can monitor operations in real time and assess past performance for maintenance planning and analysis.
- Service Oriented Architecture - Uses modern communication methods such as binary protocols and web services [26]. This setup facilitates integration with cloud platforms and industrial IoT systems.

Disadvantages

- Challenging setup - OPC UA offers extensive features like security, information modeling, and multiple communication methods, configuring it can be complex [24]. Proper deployment often requires experienced engineers.
- Greater resource requirements - Compared to lighter protocols [9], OPC UA demands more memory and processing power, which makes it harder for small embedded devices with limited hardware to run full implementations.
- Expensive deployment - The overall cost of adopting OPC UA can be high, as expenses for development, integration, licensing, training, and system setup often add up significantly.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

- Performance impact - Security mechanisms like encryption and authentication add processing overhead [15], which may slow response times in time-sensitive applications if not optimized.
- Practical interoperability issues - Despite being standardized, vendor-specific implementations can differ, leading to compatibility problems that require extra configuration or testing [23].
- Demanding to learn - Engineers moving from OPC Classic or other protocols often need considerable time to fully understand OPC UA's architecture, security framework, and information modeling approach [24].

The performances level in Table 1 are qualitative evaluations taken from the existing research. Rather than precise numerical measurements, these phrases represent how well each procedure performs in relation to others.

TABLE I.

Properties	HTTP	CoAP	MQTT	AMQT	OPC UA
transport protocol	TCP	UPD	TCP	TCP	TCP/UDP
Communication model	Request-response	Request-response	central manager "Send/Receive"	Publish-subscribe Request-response	Client-server Publish-subscribe
Latency	Medium	Low (UDP design)	lightweight	Medium	Medium
Bandwidth Usage	High	Very low	low	Medium	Medium
Reliability	High	Moderate	High (Qos levels)	High	Industrial reliability
Security support	TLS/SSL	DTLS	TLS	TLS and Authentication features	Built-in encryption and authentication
Scalability	Moderate	Moderate	High	High	High
Best application	Web/cloud APIs	Low-power sensors network	IoT telemetry/Sensor messaging	Enterprise messaging/Industrial systems	Industrial automation & complex systems

V. DISCUSSION

Different communication protocols serve different IoT requirements, and no single protocol is suitable for all application.

MQTT is well suited for large-scale systems that require continuous data transmission and efficient bandwidth usage. Its lightweight design and support for quality of service levels make it effective for scalable deployments. CoAP is more suitable where devices have limited processing power, memory, and energy resources. Its lightweight structure and UDP-based communication reduce overhead, making it efficient for low-power sensor networks.

HTTP/HTTPS remains widely used due to its compatibility with web technologies and cloud platforms. However, its higher overhead may result in increased latency and bandwidth consumption compared to lightweight IoT protocols.

AMQP and OPC UA are more suitable for enterprise and industrial applications. AMQP provides reliable and structured messaging for complex systems, while OPC UA offers strong security, interoperability, and structured data modeling. These features make them ideal for industrial environments where reliability and data integrity are critical. Overall, protocol selection should be based on system requirements such as Bandwidth usage, latency tolerance, security level, and scalability.

VI. CONCLUSION

This study compared and analyzed HTTP, MQTT, CoAP, AMQP, and OPC UA to see which ones work best in different Industrial IoT (IIoT) settings. The results show that no single protocol is always the best, each has its own strengths depending on needs like speed, bandwidth use, scalability, security, and reliability. In general, choosing the right communication protocol should be based or depend on the specific operational needs of the IIoT deployment. Every industrial setup is different some may require faster data transmission, while others may focus more on security, reliability, low power consumption, or the ability to connect to multiple devices. The protocol must match the system's goals, environment, and resource constraints to ensure smooth and efficient operation.

REFERENCES

[1] S. Jain, "Performance evaluation of MQTT and CoAP in constrained IoT environments," International Journal of Communication Systems, vol. 33, no. 14, p. e4482, 2020.

[2] S. Vashi, J. Ram, J. Modi, S. Verma and C. Prakash, "Internet of Things (IoT): A vision, architectural elements, and security issues," Procedia Computer Scienc, vol. 45, pp. 492-496, 2021.

[3] M. Iglesias-Urkia, A. Orive, M. Barcelo and A. Moran, "Analysis of CoAP, MQTT, and HTTP protocols for real-time IoT systems," IEEE Access, vol. 9, pp. 104521-104535, 2021.

[4] N. Ravi, K. Shankar and P. Kumar, "Comparative analysis of MQTT and CoAP for smart city applications," Journal of Network and Computer Applications, vol. 215, p. 103609, 2024.

[5] H. Z. J. W. Y. Chen, "Performance evaluation of HTTP3 for IoT communication systems," IEEE Internet of Things Journal, vol. 10, no. 5, p. 4123 4135, 2023.

[6] R. Kumar, D. Singh and H. Lee, "Reliability and latency analysis of MQTT and CoAP in industrial IoT environments," Future Generation Computer Systems, vol. 139, pp. 85-96, 2023.

[7] A. Pate, S. Mehta and R. Shah, "Security comparison of MQTT and CoAP protocols in IoT networks," Computer Communications, Vols. 45-57, p. 212, 2024.

[8] R. Rudevdagva and L. Riunaa, "Comparative Analysis of Real-Time Communication Protocols (Includes OPC UA, AMQP, MQTT)," International Journal of Communication and Computer Technologies, vol. 13, pp. 70-76, 2025.

[9] J. Dizdarević, F. Carpio, A. Jukan and X. Masip-Bruin, "A survey of communication protocols for Internet of Things and related challenges of fog and cloud computing integration," IEEE Access, vol. 7, pp. 86143-86162, 2019.

[10] A. Al-Fuqaha, "Internet of Things: A survey on enabling technologies, protocols, and applications," IEEE Communications Surveys & Tutorials, vol. 17, no. 4, p. 2347–2376, 2016.

[11] L. S. D. S. G. S. S. Raza, "Securing communication in the Internet of Things with CoAP," IEEE Communications Magazine, vol. 55, no. 2, pp. 36-43, 2017.

[12] "geeksforgeeks," 3 november 2025. [Online]. Available: https://www.geeksforgeeks.org/blogs/http full-form/. [Accessed 1 march 2026].

[13] M. Weyrich and C. Ebert, "Reference architectures for the Internet of Things," IEEE Software, vol. 33, no. 1, pp. 112-116, 2016.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

[14] N. Naik, "Choice of effective messaging protocols for IoT systems: MQTT, CoAP, AMQP, and HTTP," IEEE Int. Systems Engineering Symposium (ISSE), 2017.

[15] M. Bennis, M. Debbah and H. V. Poor, "Ultra-reliable and low-latency wireless communication: Tail, risk, and scale," Proceedings of the IEEE, vol. 106, no. 10, p. 1834–1853, 2018.

[16] Q. Zhao, "Researchgate," january 2017. [Online]. Available: https://www.researchgate.net/figure/The architecture-of-MQTT_fig1_313869515. [Accessed 1 march 2026].

[17] D. Thangavel, X. Ma, A. Valera, H. X. Tan and C. K. Y. Tan, "Performance evaluation of MQTT and CoAP via common middleware," IEEE Int. Conf. Communications (ICC Workshops), 2016.

[18] G. E. C. A. V.-C. M. Collina, "Introducing the QEST broker: Scaling the IoT by bridging MQTT and REST," IEEE Internet of Things Journal, vol. 5, no. 2, pp. 829-841, 2018.

[19] D. Glaroudis, "Researchgate," november 2019. [Online]. Available: https://www.researchgate.net/figure/CoAP-protocol architecture-1-3-29_fig1_337564177. [Accessed 1 march 2026].

[20] K. Mekki, E. Bajic, F. Chaxe and F. M. , "A comparative study of LPWAN technologies for large scale IoT deployment," ICT Express, vol. 5, no. 1, p. 1–7, 2019.

[21] "Researchgate," june 2017. [Online]. Available: https://www.researchgate.net/figure/AMQP communication-architecture_fig3_318910289. [Accessed 1 march 2026].

[22] K. M. S. R. P. Mahapatra, "Security and performance comparison of MQTT, CoAP, AMQP and HTTP in IoT," IEEE Int. Conf. Advanced Networks and Telecommunications Systems (ANTS), 2018.

[23] T. Sauter, "The three generations of field-level networks—Evolution and compatibility issues," IEEE Transactions on Industrial Electronics, vol. 64, no. 4, p. 3218–3227, 2017.

[24] Y. Zhang., "Performance analysis of OPC UA in industrial automation systems," IEEE Access, vol. 8, p. 51298–51307, 2020.

[25] "OPCfoundation," [Online]. Available: https://reference.opcfoundation.org/I4AAS/v100/docs /4.2. [Accessed 1 march 2026].

[26] F. Jammes and H. Smit, "Service-oriented paradigms in industrial automation," IEEE Transactions on Industrial Informatics, vol. 12, no. 2, p. 745–754, 2016.

[27] H. Derhamy, J. Eliasson and J. Delsing, "IoT interoperability—On-demand and low latency transparent multiprotocol translator," IEEE Internet of Things Journal, vol. 4, no. 5, p. 1754–1763, 2017.