Journal of Convergence in Technology and Management: Global Nexus
Web Page Info: www.jctmg.in
DOI: 10.70965/jctmgn.vol.2.issue1.16

Research Article

A STUDY ON THREE INDUSTRIALLY APPLIED LIGHTWEIGHT SECURITY PROTOCOLS: MQTT, COAP, XMPP

Rajdeep Chakraborty[1], Lokesh Vishwakarma[2]
Dept. of CSE & CSIT, Sage University, Indore, MP, India

ARTICLE INFORMATION
Article History
Received: 5th June, 2026
Revised: 3rd August
Accepted: 23rd August

Keywords:
Internet of Things (IoT), MQTT, CoAP, XMPP, Lightweight Communication Protocols, Machine-to-Machine

ABSTRACT
The Internet of Things (IoT) is rapidly growing, and lightweight communication protocols that can facilitate efficient transmittal of data in resource-limited systems are needed. However, the traditional communication approaches are prone to add high computational overhead, energy consumption, and bandwidth usage, which restricts their application in IoT systems. Such lightweight protocols as Message Queuing Telemetry Transport (MQTT), Constrained Application Protocol (CoAP), and Extensible Messaging and Presence Protocol (XMPP) have proven to be good choices for machine-to-machine communication and applications of IoT. In this paper, it is compared and analyzed from the perspective of architecture, communication mechanism, working mechanism, application, and energy consumption, latency, bandwidth consumption, scalability, security support, and other performance parameters. The results show that CoAP is the most efficient and has the least overhead in communication in constrained environments while MQTT is more reliable and scalable. XMPP, on the other hand, is interoperable and provides real-time communication, but because of its XML-based approach, it requires more resources to be used.

Cite this article as: R. Chakraborty and L. Vishwakarma, "A study on three industrially applied lightweight security protocols: MQTT, CoAP, XMPP," *Journal of Convergence in Technology and Management: Global Nexus*, vol. 2, no. 1, pp. 1–6, 2026.

Published by Learnet Publishing, Kolkata, India

1. INTRODUCTION

With the growing number of IoT devices, lightweight and efficient communication protocols are increasingly needed to transmit data across the Internet in a secure manner. Traditional communications paths which were designed for high resource environments often bring in unnecessary computational overheads, high bandwidth usage and power consumption in a low resource environment for the Internet-of-Things [1], [2]. All these restrictions have spurred a considerable number of studies on lightweight messaging protocols that support efficient machine-to-machine (M2M) communication, interoperability, and secure communication of data in heterogeneous IoT environments [3, 4].

Message Queuing Telemetry Transport (MQTT), Constrained Application Protocol (CoAP), and Extensible Messaging and Presence Protocol (XMPP) are all some of the most popular lightweight protocols. MQTT is a lightweight publish-subscribe messaging protocol, developed to minimize communication overhead for the exchange of reliable telemetry in low bandwidth and unreliable network environments [5]. CoAP is a web transfer protocol, specifically designed for constrained devices and network, which is a request/response protocol, low power consuming, low packet size [6]. XMPP is an XML-based communication protocol that is originally developed for IM and extended for IoT applications because of its ability of real-time communication, extensibility and interoperability [4]. The architecture, communication mechanisms, message overhead, scalability, energy efficiency, security implementation are vastly different for these protocols [7] and [8].

Thus, it is necessary to carry out the comparative evaluation to choose the proper protocol for lightweight and secure IoT environments. The paper is organized as follows: Literature review and motivation are discussed in Section 2. The working principles of MQTT, CoAP and XMPP are explained in Section 3. Applications and benefits of each protocol are discussed in Section 4. A comparison of the three protocols is given in Section 5. The paper is concluded in Section 6.

2. LITERATURE REVIEW AND MOTIVATION

Many research works have been conducted before on lightweight communication protocols for IoT systems to enhance the communication efficiency, resource utilization, and security in resource limited environments. Thangavel et al. [5] performed performance comparison of MQTT and CoAP in middleware and found that MQTT offers reliable communication via publish subscribe model and CoAP has low latency and lesser communication overhead in constrained networks. Naik [3] performed a study of MQTT, CoAP, AMQP and HTTP protocols, deciding that MQTT and CoAP are more appropriate for low power IoT applications as they are less bandwidth and energy intensive. Bandyopadhyay and Bhattacharyya [1] considered lightweight internet protocols for sensor web enablement and explored the various trade-offs between the two in constrained gateways deployments.

In Bellavista et al. [2] the authors discussed the convergence of mobile ad-hoc networks and wireless sensor networks in IoT urban scenarios, providing basic information about multi-protocol communication environments. Bansal and Kaur [4] compared performance of MQTT, CoAP and XMPP protocols and set a benchmark for the assessment of the performance of these protocols. Vadlamudi et al. [9] introduced a middleware API to enable heterogeneous IoT device communication, bringing up the interoperability concerns amongst protocols. To allow asynchronous communication between constrained devices, Hartke [6] formally specified the CoAP resource observation mechanism.

The recent studies have been on protocol security and performance optimization. To identify the security vulnerabilities of the authentication mechanism and the security vulnerabilities of the transport-layer protection, the security analysis of IoT communication protocols was performed by Tagliaro et al. [7] on a large scale. In order to provide the security of communication in IoT environments, the authors Gusita et al. [8] made a review on lightweight cryptographic techniques and communication improvements, highlighting efficient and lightweight protocol implementation methods. Sarkar et al. [10] assessed CoAP-based communication systems and proved its efficiency in low resource IoT environment.

While some protocols have been analysed in isolation, and some performance characteristics have been assessed, there has been little work that provides a comparative evaluation of all three protocols in terms of communication architecture, working mechanism, applications, resource utilization and lightweight suitability. Thus, the motivation of this study is to fill this gap by offering a unified comparative analysis of each protocol's effectiveness in constrained IoT environments and guidelines for selecting a protocol to be used in real-world deployments.

3. WORKING OF MQTT, COAP AND XMPP PROTOCOLS

Section 3.1, Section 3.2 and Section 3.3 illustrates the working of MQTT, CoAP and XMPP respectively.

3.1 Working of MQTT

MQTT implements a publish-subscribe communication paradigm with a broker. All messages are sent through the broker and not directly to another device! The working process of MQTT is:

- The IoT devices use TCP/IP to connect to the MQTT broker.
- Publisher – A device that publishes data messages to a particular topic.
- The published message is received and interpreted by the broker.
- Devices subscribed receive messages from the broker on their subscribed topics.
- The reliability of message delivery is managed by setting the QoS levels (set to QoS 0, QoS 1, and QoS 2).

This process is shown in figure 1, it helps to reduce network traffic, enhance the scalability and allows to communicate efficiently in low bandwidth environments [5].

Working Flow:

Figure 1: MQTT Communication Architecture

Publisher Device	→	MQTT Broker (TCP/IP)	→	Subscriber Device
Publishes to Topic		*Routes by Topic*		*Receives by Topic*

QoS Levels: 0 (At most once) | 1 (At least once) | 2 (Exactly once)

Fig. 1: MQTT Publish-Subscribe Communication Model via Broker

3.2 Working of CoAP

CoAP operates in a request / response manner like HTTP and communicates via UDP to reduce protocol overhead. The process of the work is as follows:

- A client device creates a request via REST methods (GET, POST, PUT or DELETE).
- The request is sent to the server using UDP.
- The server receives the request and sends back a response.
- Use of Confirmable (CON) messages and acknowledgments (ACK) ensures reliability.
- In addition, CoAP also provides asynchronous communication using resource observation mechanisms [6].
- The use of compact binary messages and UDP transport, as demonstrated in Figure 2, leads to lower latencies and energy consumption for constrained devices with CoAP compared to other protocols [6], [10].

Working Flow:

Figure 2: CoAP Request-Response Architecture

CoAP Client (IoT Device)	→	UDP	→	CoAP Server (Resource)
GET / POST / PUT / DELETE				*Response (2.xx / 4.xx)*

Fig. 2: CoAP RESTful Request-Response Model over UDP

3.3 Working of XMPP

XMPP is based on a client-server system architecture and is based on XML streams to exchange messages. The working mechanism consists of the following steps:

- The XMPP server is connected to client devices via TCP.
- Devices use supported security mechanisms, like SASL and TLS, to authenticate to each other.
- XML stanzas (Message / Presence / IQ) are created and sent to the server.
- The server receives and transmits the data to the receiving client or device.
- There is ongoing real-time communication and device synchronization.
- XMPP allows for real-time messaging and interoperability, as shown in Figure 3, but it means that XML processing adds communication overhead to MQTT and CoAP [4], [8].

Working Flow:

Figure 3: XMPP Client-Server Architecture

XMPP Client A (IoT Device)	→	XMPP Server (SASL/TLS Auth) XML Routing	→	XMPP Client B (Destination)
XML Stanzas (TCP)		*Message | Presence | IQ*		*Real-time Delivery*

Fig. 3: XMPP Client-Server Model with XML Stanza Exchange

4. APPLICATIONS AND ADVANTAGES OF MQTT, COAP AND XMPP PROTOCOLS

Section 4.1, Section 4.2 and Section 4.3 illustrates the applications and advantages of MQTT, CoAP and XMPP respectively.

4.1 MQTT Applications and Advantages

Applications

- Control and management of smart home automation systems.
- Industrial IoT (IIoT) monitoring and control
- Remote sensor networks
- Healthcare monitoring systems
- Smart agriculture applications

Advantages

- A light-weight protocol that has little communication overhead.
- Works well in low bandwidth and unstable networks
- Supports scalable publish-subscribe architecture:
- Multiple QoS levels are used for ensuring message delivery reliability
- Less usage power consumption for limited devices
- Secure communication over TLS/SSL support
- This section discusses the different CoAP applications and advantages.

4.2 CoAP Applications and Advantages

Applications

- CoAP is mainly used in Resource-constrained environments and low-power IoT systems [6, 10] including:
- Wireless sensor networks (WSN)
  - Smart city infrastructure
  - Environmental monitoring systems
  - Smart metering applications
  - Building automation systems
  - Energy management systems

Advantages

- Low overhead with the use of UDP communication
- RESTful architecture makes it easy to integrate the Web.
- Reduce latency and energy consumption
- The device uses fewer resources than most others.The device is efficient on constrained devices.
- Supports multicast communication
- The implementation of security is achieved with the help of DTLS.

4.3 XMPP Applications and Advantages

Applications

- Instant messaging systems
- IoT communication platforms

- Smart grid systems
- Presence-aware applications
- Real-time monitoring services

Advantages

- Allows communication in real time and asynchronously.
- Facilitates real-time and asynchronous communication.
- High interoperability, via the XML-based structure
- Extensible architecture to support custom implementations.
- High level of authentication and security
- Facilitates integration with services and discovery of devices
- Appropriate for distributed communication settings

5. COMPARATIVE STUDY OF MQTT, COAP AND XMPP PROTOCOLS

A comparative analysis of MQTT, CoAP, and XMPP is performed based on communication model, transport mechanism, security support, resource consumption, and suitability for lightweight IoT environments. These analyses are shown in Table I.

Table I: Comparative Study of MQTT, CoAP, and XMPP Protocols

Parameter	MQTT	CoAP	XMPP
Full Form	Message Queuing Telemetry Transport	Constrained Application Protocol	Extensible Messaging and Presence Protocol
Communication Model	Publish-Subscribe	Request-Response	Client-Server / Pub-Sub
Transport Protocol	TCP	UDP	TCP
Architecture	Broker-based	REST-based	XML-based Client-Server
Message Format	Lightweight Binary Header	Compact Binary Format	XML Stanzas
Default Port	1883 / 8883	5683 / 5684	5222 / 5269
Reliability Mechanism	QoS Levels (0, 1, 2)	Confirmable Messages (CON/ACK)	TCP Reliability
Security Mechanism	TLS/SSL	DTLS	TLS, SASL
Bandwidth Consumption	Low	Very Low	High
Packet Overhead	Low	Very Low	High
Energy Consumption	Low	Very Low	High
Latency	Low	Very Low	Moderate
Scalability	High	Moderate	High
Real-Time Communication	Moderate	Moderate	High
Resource Requirement	Low	Very Low	High
Suitability for Constrained Devices	High	Very High	Low
IoT Suitability	Excellent	Excellent	Moderate

6. CONCLUSION

The efficient, reliable, and scalable operation of IoT settings is heavily reliant on lightweight communication protocols. This study compared and contrasted the MQTT, CoAP and XMPP in terms of architecture, communication mechanism, resource utilization, support for security and applicability for constrained systems. From the analysis, it can be seen that MQTT with its pub/sub communication pattern and QoS support offers a good trade-off between reliability and lightweight communication, thus making it suitable to be used in scalable IoT applications [5] and [3]. The operation of CoAP is based on UDP, while its message structure is compact, resulting in the lowest communication overhead. Its bandwidth savings, low energy consumption and efficient resource usage make it the best protocol for low power IoT applications and for constrained devices [6, 10]. Compared to XMPP, although it has good interoperability, extensibility and real-time communication features, it has a higher bandwidth and computational overhead, which makes it less efficient in systems with limited resources [7] and [8]. CoAP is claimed as the lightest protocol for constrained IoT applications, while MQTT is said to offer a balanced solution for reliability and scalability. For applications that need real-time communication and interoperability, XMPP is still suitable [4], [9].

7. REFERENCES

[1] S. Bandyopadhyay and A. Bhattacharyya, "Lightweight Internet Protocols for Web Enablement of Sensors Using Constrained Gateway Devices," International Conference on Computing, Networking and Communications (ICNC), pp. 334-340, 2013.

[2] P. Bellavista, G. Cardone, A. Corradi, and L. Foschini, "Convergence of MANET and WSN in IoT Urban Scenarios," IEEE Sensors Journal, vol. 13, no. 10, pp. 3558-3567, 2013.

[3] N. Naik, "Choice of Effective Messaging Protocols for IoT Systems: MQTT, CoAP, AMQP and HTTP," IEEE International Systems Engineering Symposium (ISSE), pp. 1-7, 2017.

[4] M. Bansal and S. Kaur, "Comparative Analysis of IoT Protocols: MQTT, CoAP and XMPP," International Journal of Computer Applications, vol. 179, no. 46, pp. 20-25, 2018.

[5] D. Thangavel, X. Ma, A. Valera, H. Tan, and C. Tan, "Performance Evaluation of MQTT and CoAP via a Common Middleware," IEEE Ninth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), pp. 1-6, 2014.

[6] K. Hartke, "Observing Resources in the Constrained Application Protocol (CoAP)," IETF RFC 7641, Sep. 2015.

[7] C. Tagliaro, M. Komsic, A. Continella, K. Borgolte, and M. Lindorfer, "Large-Scale Security Analysis of Real-World Backend Deployments Speaking IoT-Focused Protocols," arXiv preprint arXiv:2405.09662, 2024.

[8] B. Gusita, A. A. Anton, C. S. Stangaciu, D. Stanescu, and M. V. Micea, "Securing IoT Edge: A Survey on Lightweight Cryptography, Anonymous Routing and Communication Protocol Enhancements," International Journal of Information Security, vol. 24, article 149, 2025.

[9] S. V. Vadlamudi, S. Krikorian, and B. Skarnes, "Implementing a Middleware API for Facilitating Heterogeneous IoT Device Communication Protocols and Data Retrieval," arXiv preprint arXiv:2312.10294, 2023.

[10] C. Sarkar, A. Das, and R. K. Jain, "Development of CoAP Protocol for Communication in Mobile Robotic Systems Using IoT Technique," Scientific Reports, vol. 15, article 9269, 2025.

[11] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," IEEE Communications Surveys & Tutorials, vol. 17, no. 4, pp. 2347-2376, 2015.

[12] J. Dizdarevic, 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," ACM Computing Surveys, vol. 51, no. 6, pp. 1-29, 2019.

[13] O. Vermesan and P. Friess, "Internet of Things: Converging Technologies for Smart Environments and Integrated Ecosystems," River Publishers, 2013.

[14] R. Minerva, A. Biru, and D. Rotondi, "Towards a Definition of the Internet of Things," IEEE Internet Initiative, vol. 1, pp. 1-86, 2015.

[15] D. Guinard and V. Trifa, "Building the Web of Things," Manning Publications, 2016.

[16] J. Granjal, E. Monteiro, and J. S. Silva, "Security for the Internet of Things: A Survey of Existing Protocols and Open Research Issues," IEEE Communications Surveys & Tutorials, vol. 17, no. 3, pp. 1294-1312, 2015.

[17] S. Li, L. Da Xu, and S. Zhao, "The Internet of Things: A Survey," Information Systems Frontiers, vol. 17, no. 2, pp. 243-259, 2015.

[18] M. Collina, G. E. Corazza, and A. Vanelli-Coralli, "Introducing the QEST Broker: Scaling the IoT by Bridging MQTT and REST," IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), pp. 36-41, 2012.

[19] Y. H. Hasan, A. Hammoodi, and A. Mansour, "Performance Analysis of IoT Application Layer Protocols: A Review," International Journal of Advanced Computer Science and Applications (IJACSA), vol. 10, no. 7, pp. 312-320, 2019.

[20] K. Mekki, E. Bajic, F. Chaxel, and F. Meyer, "A Comparative Study of LPWAN Technologies for Large-Scale IoT Deployment," ICT Express, vol. 5, no. 1, pp. 1-7, 2019.