An extended review of the application layer messaging protocol of the internet of things

Ronok Bhowmik, Md. Hasnat Riaz
Department of Computer Science and Telecommunication Engineering (CSTE), Noakhali Science and Technology University (NSTU), Noakhali, Bangladesh

Article Info

Article history:

Received Nov 15, 2022
Revised Jan 28, 2023
Accepted Feb 23, 2023

Keywords:

Advanced message queuing protocol
Constrained application protocol
Data distribution service
Internet of things
Message queuing telemetry transport
Simple text-oriented messaging protocol
Extensible messaging and presence protocol

ABSTRACT

In the internet of things (IoT), there are resource-constrained and immense heterogeneous electronic gadgets worldwide. Till now, no single IoT application layer messaging protocol is the best, nor axiomatic for every requirement. This paper exhaustively summarizes information on the messaging protocols from the available previous research sources online. Our goal is to encapsulate a simple guideline so that users can choose an optimal messaging protocol quickly according to development requirements and specifications. For this purpose, we have reviewed the literature on six enabling and evolving application layer messaging protocols used for IoT systems namely, message queuing telemetry transport (MQTT), advanced message queuing protocol (AMQP), the constrained application protocol (CoAP), extensible messaging and presence protocol (XMPP), data distribution service (DDS), and simple text-oriented messaging protocol (STOMP) in terms of some interrelated metrics. Additionally, we represented a critical analysis of the application layer messaging protocols. This study will be helpful to readers with valuable insights and guide research scholars and developers in choosing optimal application layer messaging protocols based on development specifications and requirements.

Corresponding Author:

Ronok Bhowmik
Department of Computer Science and Telecommunication Engineering (CSTE)
Noakhali Science and Technology University (NSTU)
Noakhali, Bangladesh
Email: ronokbhowmik@admin.nstu.edu.bd

1. INTRODUCTION

British entrepreneur and executive director of AUTO-ID center Kevin Ashton first introduced the word internet of things (IoT), as the title of a presentation of Proctor and Gamble in 1999 [1]. Afterward, IoT opened new windows of opportunities for technological and scientific development in every touch point we imagine. IoT systems are associated with intelligent devices, smart objects, and people [2]. Smart devices are self-configurable, self-functional, wireless-based, and can process all the work without manual or human intervention. Dependencies on the internet and internet-based services are increasing rapidly worldwide. Approximately 75.44 billion devices will be appended worldwide through the internet by 2025 [3]. IoT shoves an unbounded number of new applications in a wide range of fields like smart home systems, animal farms, productivity, supply chains, precision agriculture, environmental monitoring (low energy monitoring systems and telemetry), e-health, industrial applications, informatics, automobiles and transportation systems, high-security applications, law enforcement, defense, logistics systems, space research, entertainment systems, and wearable gadgets. IoT-related services have made everything easier than ever. IT industries named Apple, Amazon, and Google use IoT to bring innovative technological changes. In IoT, various messaging protocols are introduced based on application deployment, communication mode, suitability for applications, intrinsic nature of the devices (heterogeneity of electronic gadgets), kinds of security provided, and the nature of transmission of messages over the internet. The popularity of IoT-related devices, standards, technologies, and platforms is constantly changing and improving, and the present conditions, specifications, and requirements might not be the same in the future. So, picking up every detail (advantages and disadvantages) of the existing messaging protocols is important. Finding an optimal and cost-effective IoT messaging protocol is a mammoth task for developers. If we finalize the system design without selecting an optimal protocol and proper requirement analysis, it will cost a lot of time and money hence a project failure.

During the last fifteen years, several standardization bodies, research groups, technologists, and organizations have searched for a unique messaging protocol; unfortunately, none of them can meet all the requirements of IoT gadgets. In the context of IoT, there are hundreds of protocols of different characteristics and specifications. Some popular IoT application layer protocols are message queuing telemetry transport (MQTT) [4], constrained application protocol (CoAP) [5], extensible messaging and presence protocol (XMPP) [6], advanced message queuing protocol (AMQP) [7], data distribution service (DDS) [8], simple text-oriented messaging protocol (STOMP) [9], representational state transfer (RESTful) hypertext transfer protocol (HTTP), simple media control protocol (SMCP), lightweight local automation protocol (LLAP), simple sensor interface (SSI), lightweight machine-to-machine (LWM2M), M3DA, XMPP-IOT, ONS 2.0, simple object access protocol (SOAP), WebSocket, reactive streams, HTTP/2, and JavaScript IoT. We have noticed this work has the subsequent contribution: i) explored six well-established application layer messaging protocols of IoT systems and ii) represented a critical analysis where we compared the performance, characteristics, and behaviors of the application layer messaging protocols.

This survey paper is exhibited as follows: section 2 includes some related works in literature along with the research gap. Section 3 briefly reviews the six-application layer messaging protocols and exemplifies a critical analysis of the messaging protocols based on different parameters. In the end, in section 4, we summarize our conclusions in section 5 about this study and provide future work directions.

2. RELATED WORKS

Numerous qualitative reviews and experimental illustrations have been conducted in terms of application layer messaging protocols. As we know, IoT system standards, gadgets, specifications, and requirements are constantly changing, it is quite difficult to summarize all aspects of the system. In this paper, we have encapsulated the information from several review papers. The following section will discuss the previous works related to this review study.

Al-Fuqaha *et al.* [10] started the discussion with an overview of the IoT. The study also focused on IoT-related technologies, protocols, applications, and challenges of recent literature. In addition, the relationship between the IoT, big data analytics, cloud, and fog computing are also illustrated. Lee *et al.* [11] provided an overview of MQTT. Here, the architecture, message format, scope, and quality of service (QoS) of MQTT are described thoroughly. The authors stated that MQTT is an open standard, publish-subscribe messaging protocol which uses transmission control protocol (TCP) for transport. A comprehensive study on IoT protocols is conducted in article [12], where the authors document the recent developments of IoT protocols and standardization initiatives from the perspective of interoperability. Pathaka and Tembhurne [13] discusses the overview and the standards used in IoT. In addition, the architecture, protocols, and standards are reviewed critically. The authors also find similarities and dissimilarities between MQTT and AMQP protocols. Yugha and Chithra [14] highlighted the issues and challenges related to security and the use of IoT protocols. They also reviewed the research trends and simulation tools used for the analysis purpose of IoT application layer protocols. The goal of the survey conducted by Al-Joboury and Al-Hemiary [15] was to facilitate some guidelines for academic researchers in IoT protocols. IoT-related applications, open issues, architecture, explanation of IoT protocols, operations, functionalities, and data cloud integration are also discussed by the authors. Wytrębowicz *et al.* [16] tried to select an appropriate protocol based on specific communication requirements. Furthermore, the authors represent a comparison of the protocols based on an analysis of the protocol specifications and also provide some recommendations for proper protocol selection. Dizdarević *et al.* [17] tried to focus our attention on the implementation of fog and cloud-based IoT systems. The authors have tried to discuss interoperability, system integration, latency, power consumption, throughput, and some common features of widely used IoT protocols. Interestingly, Bayılmış *et al.* [18] emphasized the overview of lightweight communication protocols, along with their strengths and limitations. They also draw our attention to the advancement of application layer protocols for the IoT ecosystem. The authors also explained how MQTT, CoAP, and WebSocket protocols are better choices for small IoT devices. Similar to the article [13], research by Bibi *et al.* [7] represented a survey on

CoAP, MQTT, AMQP, and XMPP. The papers discuss the architecture, advantages, disadvantages, and applications of each protocol. Later, a comparative analysis is also presented.

Although the above-mentioned studies have collected a large amount of information, there is still a lack of advanced research on selecting the right IoT protocol. Surprisingly, there are inadequacies in the review of telemetry transmission, security, data integrity, and interoperability for the application layer protocols of IoT. Moreover, in our investigation, we found limited studies on DDS and STOMP protocol. Because of these shortcomings, an extended guideline has been prepared to find an optimal protocol quickly.

3. REVIEW THE APPLICATION LAYER MESSAGING PROTOCOLS FOR THE INTERNET OF THINGS

In recent years, the most ardent drift of IoT is the application layer messaging protocols and heterogeneity of electronic gadgets. One specific messaging protocol cannot satisfy all the requirements and provide a guarantee to facilitate secure, scalable, traceable, energy-optimized, time and cost-saving, and lossless communications. This section exhibits a review of the six widely accepted and emerging messaging protocols for IoT systems: MQTT, AMQP, CoAP, XMPP, DDS, and STOMP. In this section, we concisely review and compare the performance, characteristics, and behaviors of the six above-mentioned application layer messaging protocols used for IoT systems.

3.1. Message queuing telemetry transport

MQTT is a widely-used, simply designed, lightweight broker-based (server-based) message transport connectivity protocol developed in 1999 and released by Andy Stanford-Clark (IBM) and Arlen Nipper (Arcom) control systems limited [19]. It was integrated with the IBM WebSphere application server, and standardized by the organization for the advancement of structured information standards (OASIS) in 2013. OASIS aims to reduce the bandwidth requirement. It targets M2M communications and a resource-constrained environment. It follows an asynchronous publish-subscribe communication way similar to the client-server model. MQTT assumes a connection-oriented, reliable transport protocol TCP handled by transport layer security/secure sockets layer (TLS/SSL) to ensure security [20]. This protocol is suitable for sensor networks and wireless sensor networks. The architecture of MQTT contains two main components: client and broker. A client may be a publisher or subscriber, and the server is the broker. There may be more than one publisher and subscriber. Publishers/subscribers always try to make a connection to the server. Here, the server is known as the broker. A broker creates a link between physical devices and enterprise systems. Broker contains topics that are a UTF-8 string. For filtering messages to interested clients, the broker uses topics [20]. Frequently used MQTT brokers are Mosquitto, really small message broker (RSMB), MQTT.js, HiveMQ, and paho MQTT [21]. Open-source code, less message processing, lower overhead, lower network bandwidth, the ability to deal with delay or latency in the network, lower battery usage, faster response times, and straightforward implementation are the most noticeable features of MQTT. However, there are two variants of MQTT (MQTT v.3.1/v.3.1.1/v.5.0 and MQTT-SN). MQTT v.5.0. is the latest and current version [18]. MQTT-S is suitable for the non-TCP/IP stacks, and MQTT-SN is for sensory networks [22]. QoS functionalities are available for MQTT, and it has three QoS levels [11]. These QoS are denoted as QoS 0, QoS 1, and QoS 2. In QoS 0, the message is delivered at most once, in QoS 1 the message is delivered exactly once, and in QoS 2 the message is delivered at least once. MQTT isn't suitable for multicasting (one-to-many messages) [23].

In this day and age, MQTT is one of the leading open-source protocols in the IoT industry. If the circumstances are such that we have to work from remote locations where the network bandwidth and power are limited, constrained resources, and unreliable networks then we can choose MQTT. For example, in remote health monitoring scenarios (automated medical alerts), MQTT is the first choice for system development. In addition, good QoS, high security, a central broker, and a flexible subscription pattern are the notable features of MQTT. So it is clear that MQTT is the best choice for constrained environments. In most M2M communications, the extensively used protocols are the MQTT and CoAP. Slower transmit cycles than CoAP, lack of security encryption, and scalability are the drawbacks of MQTT. Lightweight applications, home automation, healthcare, social networking, enterprise-level applications, and utilities are the sphere where we are using MQTT. In addition, MQTT is used on a higher scale in various sectors in the industries such as automotive, logistics, manufacturing, smart home, consumer products, transportation, and so forth. Amazon web services, Facebook Messenger, and Microsoft Azure IoT are the sectors where MQTT is widely used nowadays [24].

3.2. Advanced message queuing protocol

AMQP is an application layer messaging protocol developed by John O'Hara at JPMorgan Chase in London, the UK, in 2003 [24] and standardized by OASIS. Like MQTT, AMQP is a broker-based message protocol similar to the client-server model [25]. In this protocol, message transmission between two nodes maintains one-to-one communication. It follows asynchronous publish-subscribe architecture. AMQP assumes connection-oriented reliable transport protocol TCP handled by TLS/SSL [26]. Compared to the REST, AMQP can send many messages per second [27]. Unlike MQTT, the AMQP broker consists of two components: exchange and queue [10]. The exchange component of the broker is responsible for performing the routing functionality by forwarding messages to the appropriate message queue. Messages are stored in a message queue until received by the receiver. This mechanism works for both end-to-end and publisher-subscriber models. There are two types of messages named bare messages and annotated messages [28]. Publishers use bare messages, and subscribers use annotated messages. Still, AMQP is not widely used and suitable for IoT sensor devices because of limited memory. As a result, its use is still quite limited within the world of IoT. Like MQTT, AMQP has three QoS levels (QoS 0, QoS 1, and QoS 2) [29]. The message is delivered at most once for QoS 0, exactly once for QoS 1, and at least once for QoS 2.

Interoperability, reliability, and trustworthiness are the striking features of the AMQP protocol [30]. It delivers acknowledgment messages of the sent messages making it crucial for banking institutions. The main interest of AMQP's development is to use it in the financial industries [30]. In that continuity, for commercial purposes in server-based analytical environments i.e., in banking industries, non-bank financial institutions (NBFI), business messaging, insurance companies, and industrial environment applications use AMQP. American banking and financial service-providing company JPMorgan use AMQP [31]. Home automation and vehicle-to-vehicle communication are the sectors where AMQP is extensively used too. Microsoft Azure service bus and Azure IoT hub support communication also use AMQP [32].

3.3. Extensible messaging and presence protocol

XMPP is an extensible markup language (XML) language-based messaging transport connectivity open-source protocol introduced in 1999 by the Jabber software foundation (JSF) [33]. It is the most heavyweight protocol. The internet engineering task force (IETF) standardized XMPP a decade ago [34]. In XMPP, there are three types of XML stanzas [35]. These are message stanza, presence stanza, and IQ stanza. Notable features of XMPP are instant messaging, real-time entertainment, telepresence, chatting, and message exchange. It targets messaging applications over the internet. It supports both publish-subscribe (asynchronous) and request-response architecture. The publish/subscribe architecture allows multicast communication. Unlike MQTT and AMQP, XMPP does not provide QoS guarantees [36]. As QoS is not guaranteed, it doesn't suit M2M communications. As we know, CoAP generates lower overhead than MQTT [37] but in XMPP, XML messages create additional overhead due to headers and tag formats that increase power consumption. Openness, scalability, extensibility, and flexibility are the noticeable features of XMPP [6]. Unlike MQTT and CoAP, XMPP consumes more power [38].

Security, real-time communication, flexibility, easy understandability, and open standard are the notable merits of XMPP. Text-based communication, no QoS, higher bandwidth, overhead, and message size are the significant demerits of XMPP. Real-time communications like instant messaging, Group chat (Google chat, Facebook chat), multi-party chat, voice and video calls, telepresence, gaming, collaboration, offline messaging, voice mailing, content syndication, and vehicle tracking are the fields where we can use the XMPP.

3.4. Constrained application protocol

CoAP is an application layer messaging protocol introduced and standardized in 2010 by IETF [39]. It targets M2M communications and is designed for constrained-resource devices [40]. The devices which don't support HTTP can use CoAP protocol. Like XMPP, CoAP supports both publish-subscribe and request-response architecture. It assumes connectionless unreliable transport protocol user datagram protocol (UDP) [41]. CoAP relies on datagram TLS (DTLS) for security purposes [42]. Instead of using topics, CoAP uses a uniform resource identifier [43]. CoAP uses GET, PUT, POST, and DELETE methods for performing the create, retrieve, update, and delete operations [44]. In CoAP, request and response messages are marked as confirmable and non-confirmable, respectively [45]. CoAP supports many-to-many communication [46].

Like MQTT, CoAP protocols are lightweight and suitable for M2M communications. If the devices have limited control functionalities, low overhead, low latency, low bandwidth, low availability, and limited RAM for M2M communication, we can choose CoAP [37]. Data authenticity and integrity, cryptographic algorithm support, confidentiality, and automatic key management are the plus points of CoAP. Supported data formats for CoAP are XML, JavaScript object notation (JSON), and concise binary object representation (CBOR) [25]. In addition to the above, limited libraries, securities, and scarcity of available solution support are the major drawbacks of CoAP. Smart home systems, mobile phones and microcontrollers, smart grids, and building automation are the sectors where we use CoAP [47].

3.5. Data distribution service

DDS is an application layer messaging protocol released and standardized by the object management group (OMG) in 2004 [48]. It is the first open interoperable middleware protocol. Mostly, it targets real-time M2M communications. It doesn't support broker based message protocol. To ensure the QoS, DDS maintains 23 levels and a variety of quality criteria [18]. Control reliability, volatility, liveliness, resource utilization, filtering and delivery, ownership, redundancy, security, durability, flexibility, urgency priority, timing deadliness, and the latency of the data are notable QoS factors of DDS. Like MQTT, DDS supports publish-subscribe and request-response architecture for real-time systems. It defines two sub-layers: data-centric publish-subscribe (DCPS), and the data-local reconstruction layer (DLRL) [32]. DCPS circulates information to the subscribers. DLRL acts as an interface to the DCPS functionalities [48]. DLRL is an optional layer that is shared among distributed objects. Distributed objects, military imaging and systems, hospital integration, cyber-physical, industrial parts of IoT, and wind firms are the sectors where we can use DDS.

3.6. Simple text-oriented messaging protocol

STOMP is a simple, text-based, lightweight, wire-format message communication protocol [49]. STOMP was introduced in 1999, and it is an open and bi-directional protocol. It is maintained by the programmers' community [50] and can perform one-to-many communications [16]. The main target of STOMP's development was to use the Apache ActiveMQ system [9]. STOMP does not have any topics or queues like MQTT. Like the other data protocols of IoT, STOMP supports the publish-subscribe architecture [51]. STOMP has more similarities to HTTP [52]. Simplicity and easy understandability are the notable features of STOMP. Compared with AMQP, STOMP is a simple and flexible protocol. The client of the STOMP protocol can communicate with almost every available STOMP message broker. This feature provides easy and widespread messaging and a wide range of language bindings, platforms, and brokers. RabbitMQ message broker uses STOMP protocols to ease and scale the deployment of modern cloud services. So far as we know, there are three versions of STOMP. These are STOMP 1.0, STOMP 1.1, and STOMP 1.2. Among these versions of the STOMP protocol, STOMP 1.2 version is the latest version. STOMP 1.2 was released worldwide on 22nd October 2012. If we have the freedom to choose any protocol in the entire network system, we can choose STOMP arbitrarily and use STOMP as a publisher and receiver. Simple message queuing applications can use STOMP. A critical analysis of the messaging protocols for IoT systems, namely, MQTT, AMQP, CoAP, XMPP, DDS, and STOMP based on several criteria to introduce their characteristics comparatively, is presented in Table 1 (see in appendix).

4. CONCLUSION

Over the past two decades, one of the dominant trends in modern technology has been IoT, and the reliance on it has been steadily increasing. The main goal of the current study is to provide an impactful guideline for choosing an appropriate IoT messaging protocol based on different specifications and requirements. This research enhances our understanding of the protocol architecture, features, and other necessary specifications of the application layer of IoT messaging protocols. The most crucial finding from the study is that no specific protocol can be called the best for all circumstances. The observations of this study suggest that we have to choose a protocol based on the type of IoT deployment, scope, project cost, power consumption, geographic size, target user groups, purpose of use, and other relevant aspects. Due to the advancement of technology in the upcoming years and severe economic recessions and energy crises worldwide, we need to be careful about power consumption. By selecting the optimal IoT protocol, we can save reasonless time, and money wastage helps in successful project completion. Hopefully, this research will enhance understanding of various gradations, usage of IoT protocols, and optimal protocol selection.

APPENDIX

Table 1. Comparison of application-level messaging protocols

Criteria	MQTT	XMPP	AMQP	CoAP	DDS	STOMP
Header size	2 bytes	Undefined/no limit	8 bytes	Minimum 4 byte	16 bytes	Implementation dependent
Maximum length	5 bytes	Unknown	Variable width	Typically, 20 Bytes and determined by the server capacity	Unknown	Implementation dependent
Encoding format	Binary (UTF-8)	Character (XML & EXI)	Binary	Binary/text	Binary	Text (HTTP like text syntax)
Overhead	Low/minimal	High/large	Low/minimal	Minimum	Unknown	High
Usage of topics	Yes	No	Yes	No	Yes	Yes
Message payload size	256 MB	Variable size (XML based payload)	32-bit	Minimum 0 and maximum 2^16-1=65535	Unknown	Unknown
Security	TLS/SASL	TLS/SSL	TLS/SASL	DTLS	TLS	Unknown
Bandwidth	Low/limited	High	Unknown	Unknown	Unknown	Unknown
Latency (ms)	0 (QoS 0) 58(QoS 1) 56 (QoS 2)	Low	0	0	Low	Unknown
Throughput (msg/sec)	75,5 (QoS 0) 14,3 (QoS 1) 7,8 (QoS 2)	Unknown	67	148,3	Unknown	Unknown
Telemetry message	Yes	No	Yes	No	No	No
Computation al capability	Depends on CPU capabilities	Unknown	Unknown	Unknown	Unknown	Unknown
Default port number	1883/8883 (TLS/SSL)	Client-5222, server-5269	TCP/UDP-5671 TCP/UDP/SCTP-5672	TCP/UDP-5683	7400	61613 TLS-61614
Memory	Limited	High	Unknown	Limitedmemory (10 KB)	Unknown	High
Power Consumption	Low	High	Unknown	Low/Reduced	Unknown	High
Microsoft Azure IoT Suite support	Yes	No	Yes	Yes	No	No
QoS Option	Yes	No	Yes	Yes	Yes	No
QoS Level	3	No	3	2	Nearly 23	No

REFERENCES

[1] K. Ashton, “That ‘internet of things’ thing,” *RFID journal*, vol. 22, no. 7, pp. 97–114, 2009.

[2] M. B. Yassein, M. Q. Shatnawi, S. Aljwarneh, and R. Al-Hatmi, “Internet of things: survey and open issues of MQTT protocol,” in *2017 International Conference on Engineering & MIS (ICEMIS)*, May 2017, pp. 1–6, doi: 10.1109/ICEMIS.2017.8273112.

[3] A. Al Hayajneh, M. Z. A. Bhuiyan, and I. McAndrew, “Improving internet of things (IoT) security with software-defined networking (SDN),” *Computers*, vol. 9, no. 1, pp. 1–14, Feb. 2020, doi: 10.3390/computers9010008.

[4] N. Q. Uy and V. H. Nam, “A comparison of AMQP and MQTT protocols for internet of things,” in *2019 6th NAFOSTED Conference on Information and Computer Science (NICS)*, Dec. 2019, pp. 292–297, doi: 10.1109/NICS48868.2019.9023812.

[5] M. A. Tariq, M. Khan, M. T. R. Khan, and D. Kim, “Enhancements and challenges in coap—a survey,” *Sensors*, vol. 20, no. 21, pp. 1–29, Nov. 2020, doi: 10.3390/s20216391.

[6] H. Wang, D. Xiong, P. Wang, and Y. Liu, “A lightweight XMPP publish/subscribe scheme for resource-constrained IoT devices,” *IEEE Access*, vol. 5, pp. 16393–16405, 2017, doi: 10.1109/ACCESS.2017.2742020.

[7] N. Bibi, F. Iqbal, S. M. Akhtar, R. Anwar, and S. Bibi, “A survey of application layer protocols of internet of things,” *International Journal of Computer Science & Network Security*, vol. 21, no. 11, pp. 301–311, 2021, doi: 10.22937/IJCSNS.2021.21.11.41.

[8] P. S. S. and B. Subramani, “Study on IoT architecture, application protocol and energy needs,” *International Journal of Scientific Research in Network Security and Communication*, vol. 8, no. 5, pp. 7–12, 2020.

[9] V. Wang, F. Salim, and P. Moskovits, “Using messaging over WebSocket with STOMP,” in *The Definitive Guide to HTML5 WebSocket*, Berkeley, CA: Apress, 2013, pp. 85–108, doi: 10.1007/978-1-4302-4741-8_5.

[10] 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, doi: 10.1109/COMST.2015.2444095.

[11] S. Lee, H. Kim, D. Hong, and H. Ju, “Correlation analysis of MQTT loss and delay according to QoS level,” in *The International Conference on Information Networking 2013 (ICOIN)*, Jan. 2013, pp. 714–717, doi: 10.1109/ICOIN.2013.6496715.

[12] R. Sutaria and R. Govindachari, “Making sense of interoperability: protocols and standardization initiatives in IoT,” in *2nd International Workshop on Computing and Networking for Internet of Things*, 2013, pp. 2–5.

[13] A. D. Pathaka and J. V. Tembhurne, “Internet of things: a survey on IoT protocols,” in *Proceedings of 3rd International Conference on Internet of Things and Connected Technologies (ICIoTCT)*, 2018, pp. 483–487, doi: 10.2139/ssrn.3168575.

[14] R. Yugha and S. Chithra, “A survey on technologies and security protocols: reference for future generation IoT,” *Journal of Network and Computer Applications*, vol. 169, pp. 1–13, Nov. 2020, doi: 10.1016/j.jnca.2020.102763.

[15] I. M. Al-Joboury and E. H. Al-Hemiary, “Internet of things (IoT): readme,” *Qalaai Zanist Scientific Journal*, vol. 2, no. 2, pp. 343–358, Apr. 2017, doi: 10.25212/lfu.qzj.2.2.35.

[16] J. Wytrębowicz, K. Cabaj, and J. Krawiec, “Messaging protocols for IoT systems—a pragmatic comparison,” *Sensors*, vol. 21, no. 20, pp. 1–32, Oct. 2021, doi: 10.3390/s21206904.

[17] 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,” *ACM Computing Surveys*, vol. 51, no. 6, pp. 1–29, Nov. 2019, doi: 10.1145/3292674.

[18] C. Bayılmış, M. A. Ebleme, Ü. Çavuşoğlu, K. Küçük, and A. Sevin, “A survey on communication protocols and performance evaluations for internet of things,” *Digital Communications and Networks*, vol. 8, no. 6, pp. 1094–1104, Dec. 2022, doi: 10.1016/j.dcan.2022.03.013.

[19] S. Bandyopadhyay and A. Bhattacharyya, “Lightweight internet protocols for web enablement of sensors using constrained gateway devices,” in *2013 International Conference on Computing, Networking and Communications (ICNC)*, Jan. 2013, pp. 334–340, doi: 10.1109/ICCNC.2013.6504105.

[20] K. Khalil, K. Elgazzar, and M. Bayoumi, “A comparative analysis on resource discovery protocols for the internet of things,” in *2018 IEEE Global Communications Conference (GLOBECOM)*, Dec. 2018, pp. 1–7, doi: 10.1109/GLOCOM.2018.8647553.

[21] Y. Upadhyay, A. Borole, and D. Dileepan, “MQTT based secured home automation system,” in *2016 Symposium on Colossal Data Analysis and Networking (CDAN)*, Mar. 2016, pp. 1–4, doi: 10.1109/CDAN.2016.7570945.

[22] S. P. Jaikar and K. R. Iyer, “A survey of messaging protocols for IoT systems,” *International Journal of Advanced in Management, Technology and Engineering Sciences*, vol. 8, no. 2, pp. 510–514, 2018.

[23] G. P. Naik and A. U. Bapat, “A brief comparative analysis on application layer protocols of internet of things: MQTT, CoAP, AMQP and HTTP,” *International Journal of Computer Science and Mobile Computing*, vol. 9, no. 9, pp. 135–141, Sep. 2020, doi: 10.47760/IJCSMC.2020.v09i09.014.

[24] N. Naik, “Choice of effective messaging protocols for IoT systems: MQTT, CoAP, AMQP and HTTP,” in *2017 IEEE International Systems Engineering Symposium (ISSE)*, Oct. 2017, pp. 1–7, doi: 10.1109/SysEng.2017.8088251.

[25] A. Chaudhary, S. K. Peddoju, and K. Kadarla, “Study of internet-of-things messaging protocols used for exchanging data with external sources,” in *2017 IEEE 14th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, Oct. 2017, pp. 666–671, doi: 10.1109/MASS.2017.85.

[26] J. Sidna, B. Amine, N. Abdallah, and H. E. Alami, “Analysis and evaluation of communication protocols for IoT applications,” in *Proceedings of the 13th International Conference on Intelligent Systems: Theories and Applications*, Sep. 2020, pp. 1–6, doi: 10.1145/3419604.3419754.

[27] J. L. Fernandes, I. C. Lopes, J. J. P. C. Rodrigues, and S. Ullah, “Performance evaluation of RESTful web services and AMQP protocol,” in *2013 Fifth International Conference on Ubiquitous and Future Networks (ICUFN)*, Jul. 2013, pp. 810–815, doi: 10.1109/ICUFN.2013.6614932.

[28] D. Glaroudis, A. Iossifides, and P. Chatzimisios, “Survey, comparison and research challenges of IoT application protocols for smart farming,” *Computer Networks*, vol. 168, pp. 1–25, Feb. 2020, doi: 10.1016/j.comnet.2019.107037.

[29] S. Appel, K. Sachs, and A. Buchmann, “Towards benchmarking of AMQP,” in *Proceedings of the Fourth ACM International Conference on Distributed Event-Based Systems*, Jul. 2010, pp. 99–100, doi: 10.1145/1827418.1827438.

[30] R. Hassan, F. Qamar, M. K. Hasan, A. H. M. Aman, and A. S. Ahmed, “Internet of things and its applications: a comprehensive survey,” *Symmetry*, vol. 12, no. 10, pp. 1–29, Oct. 2020, doi: 10.3390/sym12101674.

[31] A. Oak and R. D. Daruwala, “Assessment of message queue telemetry and transport (MQTT) protocol with symmetric encryption,” in *2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC)*, Dec. 2018, pp. 5–8, doi: 10.1109/ICSCCC.2018.8703314.

[32] E. Al-Masri *et al.*, “Investigating messaging protocols for the internet of things (IoT),” *IEEE Access*, vol. 8, pp. 94880–94911, 2020, doi: 10.1109/ACCESS.2020.2993363.

[33] X. Y. Cheng and G. Q. Dang, “Design and implementation of instant communication system based on the android platform terminal,” *Applied Mechanics and Materials*, vol. 602, pp. 3325–3328, Aug. 2014, doi: 10.4028/www.scientific.net/AMM.602-605.3325.

[34] J. Peterson, H. Tschofenig, and B. Aboba, “The role of the internet engineering task force (IETF) in improving privacy on the internet,” *W3C Workshop on Privacy for Advanced Web APIs*, pp. 1–5, 2010.

[35] A. Hornsby and R. Walsh, “From instant messaging to cloud computing, an XMPP review,” in *IEEE International Symposium on Consumer Electronics (ISCE 2010)*, Jun. 2010, pp. 1–6, doi: 10.1109/ISCE.2010.5523293.

[36] S. Schneider, “The industrial internet of things (IIoT): applications and taxonomy,” in *Internet of Things and Data Analytics Handbook*, Hoboken, NJ, USA: John Wiley & Sons, Inc., 2016, pp. 41–81, doi: 10.1002/9781119173601.ch3.

[37] D. Thangavel, X. Ma, A. Valera, H.-X. Tan, and C. K.-Y. Tan, “Performance evaluation of MQTT and CoAP via a common middleware,” in *2014 IEEE Ninth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*, Apr. 2014, pp. 1–6, doi: 10.1109/ISSNIP.2014.6827678.

[38] A. Velinov and A. Mileva, “Power consumption analysis of application layer protocols for the internet of things,” in *ICT Innovations 2016: Cognitive Functions and Next Generation ICT Systems*, 2018, pp. 193–202, doi: 10.1007/978-3-319-68855-8_19.

[39] X. Chen, “Constrained application protocol for internet of things,” *Wireless and Mobile Networking*, vol. 857, pp. 1–12, 2014.

[40] D. Schachinger and W. Kastner, “Semantic interface for machine-to-machine communication in building automation,” in *2017 IEEE 13th International Workshop on Factory Communication Systems (WFCS)*, May 2017, pp. 1–9, doi: 10.1109/WFCS.2017.7991956.

[41] A. Bhattacharjya, X. Zhong, J. Wang, and X. Li, “CoAP—application layer connection-less lightweight protocol for the internet of things (IoT) and CoAP-IPSEC security with DTLS supporting CoAP,” in *Digital Twin Technologies and Smart Cities*, Cham: Springer, 2020, pp. 151–175, doi: 10.1007/978-3-030-18732-3_9.

[42] H. G. Hamid and Z. T. Alisa, “A survey on IoT application layer protocols,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 21, no. 3, pp. 1663–1672, Mar. 2021, doi: 10.11591/ijeecs.v21.i3.pp1663-1672.

[43] D. Ugrenovic and G. Gardasevic, “CoAP protocol for Web-based monitoring in IoT healthcare applications,” in *2015 23rd Telecommunications Forum Telfor (TELFOR)*, Nov. 2015, pp. 79–82, doi: 10.1109/TELFOR.2015.7377418.

[44] M. Joshi and B. P. Kaur, “CoAP protocol for constrained networks,” *International Journal of Wireless and Microwave Technologies*, vol. 5, no. 6, pp. 1–10, Nov. 2015, doi: 10.5815/ijwmt.2015.06.01.

[45] J. Misic, V. B. Misic, and F. Banaie, “Reliable and scalable data acquisition from IoT domains,” in *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, Dec. 2017, pp. 1–6, doi: 10.1109/GLOCOM.2017.8255001.

[46] J. M. Lopez-Vega, G. Camarillo, J. Povedano-Molina, and J. M. Lopez-Soler, “RELOAD extension for data discovery and transfer in data-centric publish–subscribe environments,” *Computer Standards & Interfaces*, vol. 36, no. 1, pp. 110–121, Nov.

2013, doi: 10.1016/j.csi.2013.06.006.

[47] J. Krimmling and S. Peter, “Integration and evaluation of intrusion detection for CoAP in smart city applications,” in *2014 IEEE Conference on Communications and Network Security*, Oct. 2014, pp. 73–78, doi: 10.1109/CNS.2014.6997468.

[48] A. Balador, N. Ericsson, and Z. Bakhshi, “Communication middleware technologies for industrial distributed control systems: a literature review,” in *2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Sep. 2017, pp. 1–6, doi: 10.1109/ETFA.2017.8247730.

[49] L. Magnoni, “Modern messaging for distributed sytems,” *Journal of Physics: Conference Series*, vol. 608, no. 1, pp. 1–8, May 2015, doi: 10.1088/1742-6596/608/1/012038.

[50] “STOMP,” [Online]. Available: *stomp.github.io*. https://stomp.github.io/. access date: 31 December 2022

[51] K. Bao, I. Mauser, S. Kochanneck, H. Xu, and H. Schmeck, “A microservice architecture for the intranet of things and energy in smart buildings,” in *Proceedings of the 1st International Workshop on Mashups of Things and APIs*, Dec. 2016, pp. 1–6, doi: 10.1145/3007203.3007215.

[52] S. E. Mimouni and M. Bouhdadi, “Formal modeling of the simple text oriented messaging protocol using event-B method,” in *2015 IEEE/ACS 12th International Conference of Computer Systems and Applications (AICCSA)*, Nov. 2015, pp. 1–4, doi: 10.1109/AICCSA.2015.7507170.

BIOGRAPHIES OF AUTHORS

Ronok Bhowmik is currently employed as a Programmer in the Cyber Center at Noakhali Science and Technology University (NSTU), Noakhali, Bangladesh. He received a B.Sc. Engg. in Computer Science and Telecommunication Engineering (B.Sc. Engg. in CSTE) from the NSTU, Bangladesh in 2012. He completed an M.Sc. in Computer Science from the Department of Computer Science and Engineering of Jahangirnagar University (JU), Bangladesh in 2016. His research interests include the areas of software engineering, the internet of things (IoT), and machine learning. He can be contacted at email: ronokbhowmik@admin.nstu.edu.bd.

Md. Hasnat Riaz is working as an Assistant Professor in the Department of Computer Science and Telecommunication Engineering (CSTE), at Noakhali Science and Technology University (NSTU). He received a B.Sc. Engg. in Computer Science and Telecommunication Engineering (B.Sc. Engg. in CSTE) in 2012 and M.Sc. in Telecommunication Engineering in 2016 from the NSTU, Bangladesh. His research interests include cloud computing, mobile cloud computing, the internet of things (IoT), data science, and machine learning. He can be contacted at email: hasnat.cste@nstu.edu.bd.