Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X
IMPACT FACTOR: 7.056

IJCSMC, Vol. 9, Issue. 9, September 2020, pg.135 – 141

A Brief Comparative Analysis on Application Layer Protocols of Internet of Things: MQTT, CoAP, AMQP and HTTP

Girija P. Naik
Computer Science and Engineering Department, Goa College of Engineering, Farmagudi, Goa, India
Girijapnaik@gmail.com

Prof. A Umesh Bapat
Associate Professor in Computer Science and Engineering Department, Goa College of Engineering, Farmagudi, Goa, India
uab@gec.ac.in
DOI: 10.47760/IJCSMC.2020.v09i09.014

Abstract: The standard and real-time communication technology is a sheer prerequisite for the enhancement of Internet of Things (IoT) applications. However, the task of choosing the standard and compelling messaging protocol also known as application layer protocol is challenging, because it depends on the nature of the IoT system and its messaging requirements. In last two decades ample amount of messaging protocols have been developed and used by various organizations based on their needs. Still, not a single out of those is able to fulfil the messaging needs of all types of IoT systems. Messaging protocol is a growing perplexity for the IoT industry; thus, it is very much essential to understand the pros and cons of the widely accepted and emerging messaging protocols for IoT systems to determine their ideal scenarios. Considering the above mentioned needs, this paper tries to do an effective evaluation of the four entrenched messaging protocols MQTT, CoAP, AMQP and HTTP for IoT systems. An effort has been made to give pros and cons of each of these protocols.
Keywords: IOT systems; Messaging Protocol; MQTT; CoAP; AMQP; HTTP; Quality of Service

I. INTRODUCTION

In the era of Internet of Things (IoT), day to day things and machines are in the main role and communicate with each other. One of the major factors that determine the performance of Machine to Machine (M2M) communication is the messaging protocol specially designed for M2M communications within the IoT applications. The selection of a standard and effective messaging protocol is a challenging task for any organisation [1]. The pre-requisite for selecting an appropriate messaging protocol in IoT systems, is the better understanding of a target IoT system and its message/data sharing requirements. Unlike the Web, which uses a single standard messaging protocol HTTP, IoT cannot rely on a single protocol for all its need [2]. Consequently, there are many messaging protocols available to choose for various types of requirements of the IoT system. The future of the IoT lies on several messaging protocols and any one protocol cannot deal with all possible IoT use cases. Consequently, it is necessary to investigate the advantages and disadvantages of the widely accepted and new coming messaging protocols for IoT systems to determine their best-fit scenarios. Therefore, this paper presents a comparison of the four messaging protocols MQTT, CoAP, AMQP and HTTP. It presents the general comparison among these protocols in order to propose their characteristics analogously. Accordingly, the user can decide their relevant usage in IoT systems based on their requirements and suitability.

II. APPLICATION LAYER PROTOCOLS FOR IOT SYSTEMS

A. *MQTT* (*Message Queuing Telemetry Transport Protocol*)

One of the primeval M2M communication protocols is MQTT. It was introduced in 1999[10]. It was developed by Andy Stanford-Clark of IBM and Arlen Nipper of Arcom Control Systems Ltd (Eurotech)[10].It is a publish/subscribe messaging protocol designed for lightweight M2M communications in uneasy networks [3]. MQTT client publishes messages to an MQTT broker, which are subscribed by other clients or may be retained for the future subscription. Every message is published to an address, known as a topic [7]. Clients can subscribe to multiple topics and receives every message published to the each topic. MQTT is a binary protocol and normally requires fixed header of 2-bytes with small message payloads up to maximum size of 256 MB [6]. TCP is used as a transport protocol and TLS/SSL for security in MQTT protocol. Thus, communication between client and broker is a connection-oriented. Another really appreciating feature of MQTT is its three levels of Quality of Service (QoS) for reliable delivery of messages [3]. MQTT is most suitable for large networks of small devices that need to be monitored or controlled from a back-end server on the Internet. It is neither designed for device-to-device transfer nor for multicast data to many receivers [7].

Advantages of MQTT:

- Reliability of messages is provided by supporting QoS levels.
- Utilizes bandwidth through packet skeptic effectively. The data may contain binary or text.[11]
- Publish/Subscribe mechanism has capabilities such as one-to-one, many-to-many or one-to-none. Also, this mechanism provides bi-directional communication. .[11]
- Easy methods are used for communication.
- Asynchronous communications among nodes. Messages can publish/subscribe anytime. .[11]

Disadvantages of MQTT:

- It uses TCP/IP and TCP requires more communication capabilities unlike UDP. .[11]
- There is a limited capacity for communication for Broker.
- All nodes are connected to Broker. Therefore, the communication collapses when the broker is a failure. .[11]

B. *CoAP (Constrained Application Protocol)*

CoAP is a lightweight M2M protocol from the IETF CoRE (Constrained RESTful Environments) Working Group. CoAP supports both request/response and resource/observe (a variant of publish/subscribe) architecture [3]. Unlike MQTT, CoAP uses Universal Resource Identifier (URI) instead of topics [6]. Data to the URI is published by the publisher and subscriber subscribes to a particular resource indicated by the URI. When a publisher publishes new data to the URI, then all the subscribers are notified about the new value as indicated by the URI. CoAP is a binary protocol and normally requires fixed header of 4-bytes with small message payloads up to maximum size dependent on the web server or the programming technology [6]. CoAP uses UDP as a transport protocol and DTLS for security [8]. Thus, communication through connectionless datagrams with less reliability takes place between clients and servers. However, it uses “confirmable” or “non-confirmable” messages to provide two different levels of QoS. Where, confirmable messages must be acknowledged by the receiver with an ACK packet and nonconfirmable messages are not. CoAP supports content negotiation to express a preferred representation of a resource; this allows client and server to evolve independently, adding new representations without affecting each other.

Advantages of CoAP:

- Sends small packets with UDP layer. Thus provides fast communication.
- Provides Asynchronous communication. .[11]
- Many-to-many communication is supported.
- Datagram Transport Layer Security (DTLS) provides integrity, security, and privacy by authorizing encrypting and securing. .[11]
- For constrained devices it is a very efficient option. .[11]

Disadvantages of CoAP:

- Messages are unreliable. 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. .[11]
- It is still standardizing. It is selected the most unstandardized protocol among other protocols. .[11]

C. *AMQP (Advanced Message Queuing Protocol)*

AMQP is a lightweight M2M protocol, and was developed by John O’Hara at JPMorgan Chase in London, UK in 2003[10]. Advanced Message Queuing Protocol is an open standard protocol designed to support messaging over middle-ware. AMQP created a functional interoperability between the client and the messaging middle-ware[10]. The model consists of a set of components that route the messages within the broker service and a network wire live protocol lets client application to communicate with server and interact with the AMQ model. AMQP communication system requires that either the publisher or consumer creates an “exchange” with a given name and then broadcasts that name. Publishers and consumers use the name of this exchange to discover each other. Subsequently, a consumer creates a “queue” and attaches it to the exchange at the same time. Messages received by the exchange have to be matched to the queue via a process called “binding”. [10]The protocol is used in distributed application and it includes point- to-point, publish, subscribe, fan-out and request-response messaging system. AMQP does not store messages, instead, the messages are routed to queues on behalf of recipient. [12]

The Advanced Message Queuing Protocol was designed to provide features like open source, standardization, reliability, interoperability and security. It helps in connecting the organization, time, space and technologies.[12] The protocol is binary, with features like negotiation, multichannel, portability, efficiency and asynchronous messaging. It is commonly split into two layers, namely, a functional layer and a transport layer. The functional layer helps in defining the commands for functioning on the part of the application, whereas the transport layer helps in carrying the different techniques such as framing, channel multiplexing, data representation, etc., between the server and the application. AMQP is a binary protocol and normally requires fixed header of 8-bytes with small message payloads up to maximum size dependent on the broker/server or the programming technology [14], [15]. AMQP uses TCP as a default transport protocol and TLS/SSL and SASL for security [13]. The Advanced Message Queuing protocol provides some key features that are beneficial for organizations as well as for applications. Rapid and guaranteed message deliveries, as well as reliability and message acknowledgments, are the main features of the protocol. These abilities help in the distribution of messages in a multi-client environment, in the delegation of time-consuming tasks and in making a server tackle immediate requests faster. The protocol also has the capability to globally share and monitor updates and also to enable communication between different systems that are connected. Another advantage of the protocol is full asynchronous functionality for systems as well as improved reliability and better uptime with regard to application deployments[12]

Advantages of AMQP:

- It uses QoS and hence ensures safe passage of important data.[16]
- AMQP uses already established publish/subscribe architecture for data sharing as used by MQTT protocol.
- It ensures interoperability as it uses wire level protocol which sends data as stream of bytes.
- It offers simpler peer to peer communication along with intermediaries.
- The protocol has space to evolve to work with different standards. [16]
- It offers secured connection to users using SSL protocol.

Disadvantages of AMQP:

- It is not backward compatible with old versions.
- It is not as simple as HTTP or any other wire protocols. [16]
- It requires higher bandwidth.
- Resource discovery is not supported. [16]

D. *HTTP (Hyper Text Transport Protocol)*

HTTP is predominantly a web messaging protocol, which was originally developed by Tim Berners-Lee[10]. Later, it was developed by IETF and W3C jointly and first published as a standard protocol in 1997 [13]. HTTP supports request/response RESTful Web architecture. Instead of topics, as compared to CoAP, HTTP uses Universal Resource Identifier (URI). Server sends data through the URI and client receives data through particular URI. HTTP is a text-based protocol and it does not define the size of header and message payloads rather it depend on the web server or the programming technology. HTTP uses TCP as a default transport protocol and TLS/SSL for security [9]. Thus, communication between client and server is a connection-oriented. It does not explicitly define QoS and requires additional support for it. HTTP is a globally accepted web messaging standard offers several features such as persistent connections, request pipelining, and chunked transfer encoding [4], [5], [9].

Advantages of AMQP:

- It offers lower CPU and memory usage due to less simultaneous connections.
- It offers reduced network congestion as there are fewer TCP connections. [16]
- It reports errors without closing the TCP connection. [16]

Disadvantages of AMQP:

- It can be used for point to point connection.
- It is too verbose. [16]
- It does not offer reliable exchange (without retry logic). [16]

III. COMPARATIVE ANALYSIS OF MESSAGING PROTOCOLS FOR IOT SYSTEMS: HTTP, COAP, AMQP AND MQTT[10]

Criteria	MQTT	CoAP	AMQP	HTTP
1. Year	1999	2010	2003	1997
2. Architecture	Client/Broker	Client/Server or Client/Broker	Client/Broker or Client/Server	Client/Server
3. Abstraction	Publish/Subscribe	Request/Response or Publish/Subscribe	Publish/Subscribe or Request/Response	Request/Response
4. Header Size	2 Byte	4 Byte	8 Byte	Undefined
5. Message Size	Small and Undefined (up to 256 MB maximum size)	Small and Undefined (normally small to fit in single IP datagram)	Negotiable and Undefined	Large and Undefined (depends on the web server or the programming technology)
6. Semantics/ Methods	Connect, Disconnect, Publish, Subscribe, Unsubscribe, Close	Get, Post, Put, Delete	Consume, Deliver, Publish, Get, Select, Ack, Delete, Nack, Recover, Reject, Open, Close	Get, Post, Head, Put, Patch, Options, Connect, Delete
7. Cache and Proxy Support	Partial	Yes	Yes	Yes
8. Quality of Service (QoS)/ Reliability	QoS 0 - At most once (Fire-and-Forget), QoS 1 - At least once, QoS 2 - Exactly once	Confirmable Message (similar to At most once) or Non-confirmable Message (similar to At least once)	Settle Format (similar to At most once) or Unsettle Format (similar to At least once)	Limited (via Transport Protocol - TCP)
9. Standards	OASIS, Eclipse Foundations	IETF, Eclipse Foundation	OASIS, ISO/IEC	IETF and W3C
10. Transport Protocol	TCP (MQTT-SN can use UDP)	UDP, SCTP	TCP, SCTP	TCP

11. Security	TLS/SSL	DTLS, IPSec	TLS/SSL, IPSec, SASL	TLS/SSL
12. Default Port	1883/ 8883 (TLS/SSL)	5683 (UDP Port)/ 5684 (DLTS)	5671 (TLS/SSL), 5672	80/ 443 (TLS/SSL)
13. Encoding Format	Binary	Binary	Binary	Text
14. Licensing Model	Open Source	Open Source	Open Source	Free
15.Organisational Support	IBM, Facebook, Eurotech, Cisco, Red Hat, Software AG, Tibco, ITSO, M2Mi, Amazon Web Services (AWS), InduSoft, Fiorano	Large Web Community Support, Cisco, Contiki, Erika, IoTivity	Microsoft , JP Morgan, Bank of America, Barclays, Goldman Sachs, Credit Suisse	Global Web Protocol Standard

Note: Reprinted from "Choice of Effective Messaging Protocols for IoT Systems: MQTT, CoAP, AMQP and HTTP," by "Nitin Naik Defence School of Communications and Information Systems Ministry of Defence, United Kingdom", Email: nitin.naik100@mod.gov.uk

IV. CONCLUSION

In this paper we have tried to present a fair evaluation of the four widely accepted and rising messaging protocols for IoT systems: MQTT, CoAP, AMQP and HTTP. An overall comparison among these protocols is done by making a study of various published materials. The Pros and Cons of each of these protocols have been presented. This study might help the user to get a gist of each of these protocols.

REFERENCES

[1]. N. Naik, P. Jenkins, P. Davies, and D. Newell, "Native web communication protocols and their effects on the performance of web services and systems," in 16th IEEE International Conference on Computer and Information Technology (CIT). IEEE, 2016, pp. 219–225.

[2]. N. Naik and P. Jenkins, "Web protocols and challenges of web latency in the web of things," in 2016 Eighth International Conference on Ubiquitous and Future Networks (ICUFN). IEEE, 2016, pp. 845–850.

[3]. S. Bandyopadhyay and A. Bhattacharyya, "Lightweight internet protocols for web enablement of sensors using constrained gateway devices," in Computing, Networking and Communications (ICNC), 2013 International Conference on. IEEE, 2013, pp. 334–340.

[4]. V. Gazis, M. Gortz, M. Huber, A. Leonardi, K. Mathioudakis, A. Wiesmaier, F. Zeiger, and E. Vasilomanolakis, "A survey of technologies for the internet of things," in 2015 IEEE International Wireless Communications and Mobile Computing Conference, 2015, pp. 1090–1095.

[5]. V. Karagiannis, P. Chatzimisios, F. Vazquez-Gallego, and J. AlonsoZarate, "A survey on application layer protocols for the internet of things," Transaction on IoT and Cloud Computing, vol. 3, no. 1, pp. 11–17, 2015.

[6]. 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 Intelligent Sensors, Sensor Networks and Information Processing, 2014 IEEE Ninth International Conference on. IEEE, 2014, pp. 1–6.

[7]. T. Jaffey. (2014, February) MQTT and CoAP, IoT protocols. [Online]. Available: https://eclipse.org/community/eclipse newsletter/ 2014/february/article2.php

[8]. A. Ludovici, P. Moreno, and A. Calveras, "TinyCoAP: a novel constrained application protocol (CoAP) implementation for embedding RESTful web services in wireless sensor networks based on tinyos," Journal of Sensor and Actuator Networks, vol. 2, no. 2, pp. 288–315, 2013.

[9]. I. Grigorik, "Making the web faster with HTTP 2.0," Communications of the ACM, vol. 56, no. 12, pp. 42–49, 2013.

[10]."Choice of Effective Messaging Protocols for IoT Systems: MQTT, CoAP, AMQP and HTTP," by

"Nitin Naik Defence School of Communications and Information Systems Ministry of Defence, United Kingdom", Email: nitin.naik100@mod.gov.uk

[11]. Comparative Analysis of IoT Communication Protocols, Burak H. Çorak1 , Feyza Y. Okay1, Metehan Güzel1, Şahin Murt2, Suat Ozdemir1 1Department of Computer Engineering, Gazi University, Ankara, Turkey 2AIR Telekomünikasyon Çözümleri San. Tic. A.Ş. Ankara, Turkey, 978-1-5386-3779-1/18/$31.00 ©2018 IEEE

[12]. https://cio-wiki.org/wiki/Advanced_Message_Queuing_Protocol_(AMQP)

[13]. N. S. Han, "Semantic service provisioning for 6LoWPAN: powering internet of things applications on web," Ph.D. dissertation, Institut National des Tel´ ecommunications, 2015. ´

[14]. J. E. Luzuriaga, M. Perez, P. Boronat, J. C. Cano, C. Calafate, and P. Manzoni, "A comparative evaluation of AMQP and MQTT protocols over unstable and mobile networks," in 12th Annual IEEE Consumer Communications and Networking Conference, 2015, pp. 931–936.

[15]. G. Marsh, A. P. Sampat, S. Potluri, and D. K. Panda, "Scaling advanced message queuing protocol (AMQP) architecture with broker federation and infiniband," Ohio State University, Tech. Rep. OSU-CISRC-5/09- TR17, 2008.

[16]. https://www.rfwireless-world.com/Terminology

