A Comparative Reconnaissance Review on IoT Application Layer

Krishna Suchak[a]*, Rajan Patel[b], Maitri Patel[c]

[a,b,c] *Gandhinagar Institute of Technology, Khatraj-Kalol Road, Gandhinagar-382721, India*

Abstract

Over billions of ‘things’ got connected to the internet for the past few years. Devices are connected and pass the data. To provide effective communication between devices, the application protocol of IoT is selected. By connecting to IoT gateway, IoT devices share the sensor data. These data are passed through the network and it creates traffic. To reduce internet traffic, the appropriate protocol is selected which also improves reliability. The goal of the Internet of things (IoT) is to ensure effective communication between different objects. For providing services the application layer is responsible and provides a set of protocols for message passing at the application level. This survey shows different protocols that are used in IoT, to affirm a reliable communication between objects and things.

Keywords: IoT, Application layer, Protocol, MQTT, COAP, XMPP, AMQP.

1. Introduction

Billions of devices nowadays are interconnected and communicate to receive or send data. This is done by different protocols of the application layer in the Internet of Things (IoT). The evolvement of IoT shows that it is not a simple network of computers, but a network of various devices [1]. According to survey, in 2010, Devices connected to IoT has surpassed the number of total human population on earth [2]. With IoT, objects are connected and become smart. This is used in different fields like smart home, smart agriculture, smart healthcare, etc. Here objects work smart by transferring data [3,4]. IoT supports a large range of applications with contravention requirements and components [5]. The basic IoT smart grid consists of 3 layers: perception layer, network layer, and application layer [6].

Out of 3 layers of IoT Application layer is what the user will interact with. The application layer provides specific services to users through analysis and processing data [7]. The application layer interacts directly with the end-user which consists of applications each with its own application layer protocols.

The first section, the introduction gives an overview of IoT and its application layer protocols. The second section describes the application layer protocols and its architecture: Message queue telemetry transport (MQTT), Constrained Application Protocol (CoAP), Extensible Messaging and Presence Protocol (XMPP), and Advanced Message Queuing Protocol (AMQP). The third section gives the comparison of IoT application layer protocols and is finally concluded in the last section.

2. IoT Application Layer Protocols

IoT leads to various innovations of frameworks that detect and react without human interactions, when the fire is detected it sends the instant alert message to registered users [8]. Through the Application layer user interacts. The application layer is the interface between end devices and the networks. It provides high-quality services to meet users’ requirements.

2.1. MQTT

The MQTT is a publish/subscribe model. Publisher and subscriber can switch roles depending on requirements and objectives. MQTT is suited for a constrained environment like low power, limited memory, and limited bandwidth as clients do not have to request updates [9]. It is suited for IoT applications and runs over TCP/IP [10,11]. MQTT protocol is simple and does not need high CPU and memory usage, it is a lightweight protocol. Fig.1 shows MQTT Architecture where there is an MQTT broker who works as middleware to producers and subscribers. Furthermore, MQTT does not require a request on message update which saves battery life and bandwidth [12]. MQTT is very much useful in communicating with low-power devices [13].

MQTT ensures reliability by providing the option of three QoS levels:

* Krishna Suchak
E-mail address: krishnasuchak18@gmail.com

QoS0: (At most once) it sends messages only once. The publisher sends data to the broker, in response it doesn't wait for an acknowledgment (ACK) from the broker. If the data sent by the publisher is not received by the broker, it is lost as there are no retransmissions in this QoS

QoS1: (At least once) to avoid the earlier problem of data, the publisher waits for ACK (APUBACK) from the broker. If the ACK is not received after a predefined time interval, data is retransmitted. This profile achieves reliability but increases the overhead.

QoS2: (Exactly once) in this, the publisher sends data to the broker and wait for Publish Receive (PUBREC) message back. It discards the reference to published data and Publishes Released (PUBREL) to the broker when PUBREC is received [14]. The same procedure is followed by the broker. When both publisher and broker perform their tasks, it ensures successful message delivery.

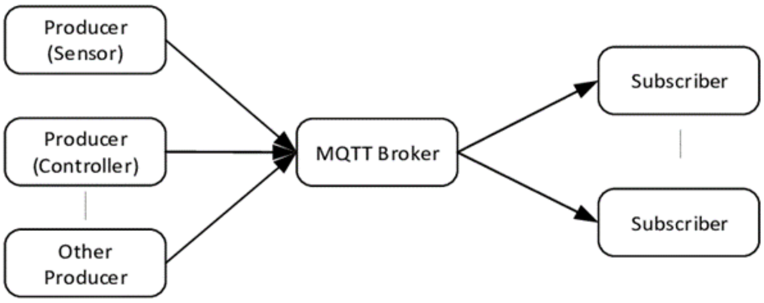

Fig. 1. MQTT Architecture [15]

2.2 COAP

The COAP is a synchronous request/response application layer protocol. Which aims to target constrained-recourse devices. The reason for bad packet delivery and high overhead is the constrained environment. CoAP was designed by the Internet Engineering Task Force (IETF). IETF is highly interested in the machine-to-machine (m2m) applications and the automation of systems to lower overhead, increase packet delivery, and make tasks simple, by using a simple interface with HTTP. To support a large number of users and provide a better performance, Publish /subscribe architecture is used [16]. Fig. 2 shows COAP Architecture. COAP incurs lower message size and overhead comparatively [17].

Following are two layers of COAP:

1) Messaging layer: The first layer, messaging layer aims to achieve reliability based on UDP [8].

2) Request/Response layer: The request/response layer aims to act the interactions and communication [8].

Types of massages in CoAP are:

A. Conformable Message: This type of message guarantees reliable communication by using the acknowledgment method; when a message arrives at its destination it returns an acknowledgment or reset message.

B. Non-conformable: Here there is no need for an acknowledgment message.

C. Acknowledgment Message: This message indicates that a conformable message has arrived.

D. Reset Message: Reset message shows that message was received (confirmable message or non-confirmable message), but due to some reason it was not executed properly. The main reason for this is when the receiving node has rebooted and has forgotten some state that would be required to interpret the message [18].

E. Piggybacked Response: As soon as the message of acknowledgment is received, the Receiver responds directly to it.

F. Separate Response: Here, receiver will respond in separate message to acknowledgement message.

CoAP is simple and consumes less CPU and memory. On the other hand, though, it is known for its high latency, bad packet delivery, and its inability to be used on complex data types [19].

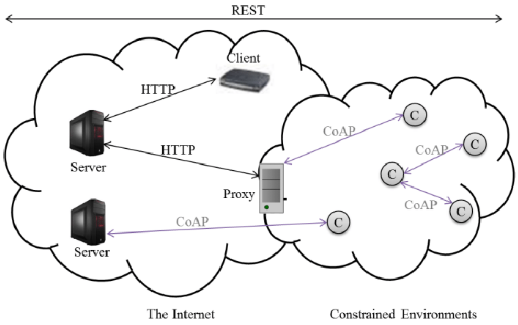

Fig. 2. COAP Architecture [20]

2.3 XMPP

The Extensible Messaging and Presence Protocol (XMPP) was designed for chatting and message exchanging. It is a well-proven protocol that has been used widely all over the Internet. As this protocol is older compared to other protocols, it does not provide the required services for latest data applications. For this reason, last year, Google stopped supporting the XMPP standard due to the lack of worldwide support.

XMPP runs over TCP which provides publish/subscribe and also request/response messaging systems. Publish subscribe is asynchronous whereas request response is synchronous. It is designed for near real-time communications and thus, it supports a small message footprint and low latency message exchange [21]. Fig. 3 shows XMPP Architecture. XMPP is extensible and allows the specification of XMPP Extension Protocols (XEP) by which its functionality is increased. It is also believed that XMPP is one of the best protocols for new era of cloud computing [22].

XMPP protocol uses XML for text communications, this may cause network traffic overhead, but it could be solved by compressing XML using EXI [23].

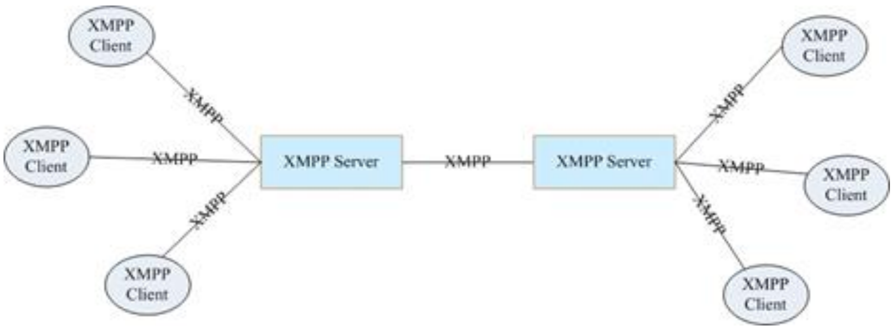

Fig. 3. XMPP Architecture [24]

2.4 AMQP

AMQP is a publish/subscribe model, depending on a coherent and authentic messaging queue [8]. The use of this approach makes the AMQP protocol easy to use and manage [25]. Applications that belongs to the AMQP protocol can exchange message from one to another. AMQP protocol focuses on achieving high reliability, security, and performance [26]. Because of its wide range of services related to messaging AMQP is the preferred choice for business [27].

There are two components of publish/subscribe approach of AMQP, that is Exchange queue and message queue,

1) The exchange queue is responsible for message routing.

2) The message queue keeps storing messages until they are sent to the receiver.

Publish/subscribe approach of AMQP consists of two components: There is a specific process with a set of rules to exchange messages between exchange components and message queues [14]. Fig.4 AMQP architecture which shows broker working between Publishers and Subscribers where they can publish/subscribe data.

AMQP protocol is Interoperable, highly extendable in different platforms and environments with good ability which can also support industrial applications. In addition, AMQP also offers more aspects considering security [28].

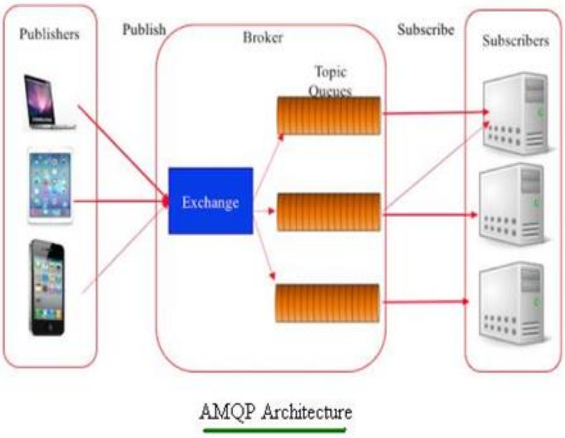

Fig. 4. AMQP Architecture [29]

Table 1. Application of Protocols

Protocols	Application name	Current version	Link	Released year
MQTT	Facebook Messanger	347.0.0.8.115 (299411680)	https://www.facebook.com/messenger/	August 9, 2011
COAP	Smart energy grid	Release 4.0	https://www.nist.gov/el/smart-grid	2007
XMPP	Firebase Cloud Messaging	CLI v9.12.1.	https://firebase.google.com/docs/cloud-messaging	October 21, 2014
AMQP	RabbitMQ	3.9.13	https://www.rabbitmq.com/	July 1, 2007

3. Comparison of IoT protocols

Table 2. Comparison of Protocols

Protocols	Types of Services provided	Architecture	Security	Header Size	Encoding Format
MQTT	TCP	Publish/Subscribe	TLS/SSL	2 bytes	Binary
COAP	UDP	Request/Response	DTLS	4 bytes	Binary
XMPP	TCP	Request/Response Publish/Subscribe	TLS/SSL	Very large with no limits (varies on data size)	XML
AMQP	TCP	Publish/Subscribe	TLS/SSL	8 bytes	Binary

4. Conclusion

This paper briefly discusses different protocols of the application layer. This comparison helps the researcher to select protocols based on the requirements. The main objective of this paper was to analyze the types of services, architecture, and security provided by each protocol. Further work can be carried out by implementing all these protocols which can give a more accurate idea of selecting protocol based on the requirements. We can also aim to implement a server that use multiple protocols and can provide the best result in terms of overall performance.

References

1. H. Hejazi, H. Rajab, T. Cinkler and L. Lengyel, "Survey of platforms for massive IoT," 2018 IEEE International Conference on Future IoT Technologies (Future IoT), 2018, pp. 1-8, doi: 10.1109/FIOT.2018.8325598.
2. D. Evans, "The internet of things: How the next evolution of the internet is changing everything," CISCO White Paper, 2011.
3. P. Datta and B. Sharma, "A survey on IoT architectures, protocols, security and smart city based applications," 2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT), 2017, pp. 1-5, doi: 10.1109/ICCCNT.2017.8203943.
4. M. H. Miraz, M. Ali, P. S. Excell, and R. Picking, "A Review on Internet of Things (IoT), Internet of Everything (IoE) and Internet of Nano Things (IoNT)", in 2015 Internet Technologies and Applications (ITA), pp. 219– 224, Sep. 2015, DOI: 10.1109/ITechA.2015.7317398.
5. Zhihong Yang, Yingzhao Yue, Yu Yang, Yufeng Peng, Xiaobo Wang and Wenji Liu, "Study and application on the architecture and key technologies for IOT," 2011 International Conference on Multimedia Technology, 2011, pp. 747-751, doi: 10.1109/ICMT.2011.6002149.
6. N. Naik, "Choice of effective messaging protocols for IoT systems: MQTT, CoAP, AMQP and HTTP," 2017 IEEE International Systems Engineering Symposium (ISSE), 2017, pp. 1-7, doi: 10.1109/SysEng.2017.8088251.
7. L. Šikić et al., "A Comparison of Application Layer Communication Protocols in IoT-enabled Smart Grid," 2020 International Symposium ELMAR, 2020, pp. 83-86, doi: 10.1109/ELMAR49956.2020.9219030.
8. M. Sheth, A. Trivedi, K. Suchak, K. Parmar and D. Jetpariya, "Inventive Fire Detection utilizing Raspberry Pi for New Age Home of Smart Cities," 2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT), 2020, pp. 724-728, doi: 10.1109/ICSSIT48917.2020.9214108.
9. M. B. Yassein, M. Q. Shatnawi and D. Al-zoubi, "Application layer protocols for the Internet of Things: A survey," 2016 International Conference on Engineering & MIS (ICEMIS), 2016, pp. 1-4, doi: 10.1109/ICEMIS.2016.7745303.
10. Banks, A., & Gupta, R. (2014). MQTT Version 3.1. 1. OASIS Standard.
11. A. Chaudhary, S. K. Peddoju and K. Kadarla, "Study of Internet-of-Things Messaging Protocols Used for Exchanging Data with External Sources," 2017 IEEE 14th International Conference on Mobile Ad Hoc and Sensor Systems (MASS), 2017, pp. 666-671, doi: 10.1109/MASS.2017.85.Shriram K Vasudevan, Abhishek S Nagarajan, RMD Sundaram, 2018. Internet of Things, Wiley.
12. H. W. Chen and F. J. Lin, "Converging MQTT Resources in ETSI Standards Based M2M Platform," 2014 IEEE International Conference on Internet of Things (iThings), and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom), 2014, pp. 292-295, doi: 10.1109/iThings.2014.52.
13. OASIS, Standard OASIS Advanced Message Queuing Protocol AMQP Version 1.0, 2014.
14. Standard, O. A. S. I. S. Oasis advanced message queuing protocol (amqp) version 1.0., 2012. URL http://docs. Oasis-open. org/amqp/core/v1. 0/os/amqp-core-complete-v1. 0-os. Pdf.
15. YANG, Wen & Haider, Naeem & ZOU, Jian-hong & Zhao, Qianchuan. (2017). Industrial Big Data Platform Based on Open Source Software. 10.2991/cnct-16.2017.90.
16. Oh, S., Kim, J. H., & Fox, G. (2010). Real-time performance analysis for publish/subscribe systems. Future Generation Computer Systems, 26 (3), 318-323.
17. 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.
18. M. S. Meera and S. N. Rao, "Comparative Analysis of IoT protocols for a Marine IoT System," 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2018, pp. 2049-2053, doi: 10.1109/ICACCI.2018.8554906.
19. Talaminos-Barroso, A., Estudillo-Valderrama, M. A., Roa, L. M.rrrr, Reina-Tosina, J., & Ortega-Ruiz, F. (2016). A Machine-to-Machine protocol benchmark for eHealth applications–Use case: Respiratory rehabilitation. Computer Methods and Programs in Biomedicine, 129, 1-11.
20. Ouakasse, Fathia & Rakrak, Said. (2019). An Improved Adaptive CoAP Congestion Control Algorithm. International Journal of Online and Biomedical Engineering (iJOE). 15. 96. 10.3991/ijoe.v15i03.9122.
21. Sven Bendel, Thomas pringer, Daniel Schuster, Alexander Schill, Ralf Ackermann, Michael Ameling, A Service Infrastructure for the Internet of Things based on XMPP, IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops), 18-22 March 2013, pp. 385-388.
22. V. Karagiannis, P. Chatzimisios, F. Vazquez-Gallego, and J. AlonsoZarate, "A Survey on Application Layer Protocols for the Internet of Things," Trans. IoT Cloud Comput., vol. 3, no. 1, pp. 11–17, 2015.
23. Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., & Ayyash, M. (2015). Internet of things: A survey on enabling technologies, protocols, and applications. Communications Surveys & Tutorials, IEEE, 17(4), 2347-2376.

24. Latvakoski, Juhani & Iivari, Antti & Vitic, Paul & Jubeh, Bashar & Alaya, Mahdi & Monteil, Thierry & Lopez, Yoann & Talavera, Guillermo & Gonzalez, Javier & Granqvist, Niclas & Kellil, Mounir & Ganem, Herve & Väisänen, Teemu. (2014). A Survey on M2M Service Networks. Computers. 3. 10.3390/computers3040130.
25. Bloebaum, T. H., & Johnsen, F. T. (2015, October). Evaluating publish/subscribe approaches for use in tactical broadband networks. In Military Communications Conference, MILCOM 2015-2015 IEEE (pp. 605-610). IEEE.
26. Fernandes, J. L., Lopes, I. C., Rodrigues, J. J., & Ullah, S. (2013, July). Performance evaluation of RESTful web services and AMQP protocol. In Ubiquitous and Future Networks (ICUFN), 2013 Fifth International Conference on (pp. 810-815). IEEE.
27. Hornsby, A., “From Instant Messaging to Cloud Computing, an XMPP Review”, IEEE 14th International Symposium on Consumer Electronics (ISCE), 7-10 June 2010, pp 1 – 6.
28. 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," 2015 12th Annual IEEE Consumer Communications and Networking Conference (CCNC), 2015, pp. 931-936, doi: 10.1109/CCNC.2015.7158101.
29. Karpenko, Anastasiia. (2017). Practical Use of O-MI/O-DF messaging standards in mobile application for IoT. Creating an open system for smart EV charging.. 10.13140/RG.2.2.36333.08163.