iJournals: International Journal of Software & Hardware Research in Engineering (IJSHRE)
ISSN-2347-4890
Volume 9 Issue 5 May 2021

Exchanging Data Using MQTT Protocol In Arduino

Arefa Shafique Shaikh[1]; Narendra Joshi[2]

P.G. Student, Department of Computer Science Engineering, Sandip University Nashik, Maharashtra, India[1]; Assistant Professor, Department of Computer Science Engineering, Sandip University Nashik, Maharashtra, India[2]

affishaikh72@gmail.com[1]; narendra.joshi@sandipuniversity.edu.in[2]

ABSTRACT

In the coming years, sensors will likely grow in every aspect of our lives. Several activities explain how the Internet of Things (IoT) will have an impact on almost all aspect of our lives and why security is at the top of the list of IoT challenges. Device to Device communications (D2D) in IoT are forecast and another major concern within the use of IoT is to make sure device security, D2D connectivity and high quality data. Therefore, a proper communication protocol is required to fix this issues. To address this, we purpose the use of Message Queue Telemetry Transport(MQTT)protocol to transfer data between devices, as it is more secured. MQTT (Message Queuing Telemetry Transport) is a publish/subscribe messaging protocol which works on top of the TCP/IP protocol. The key feature of MQTT is its light weight, adds flexible authentication and bandwidth efficiency. The result of this study is transferring high quality data securely using MQTT protocol.

Keywords: MQTT, TLS, AES, DES, HTTP, IoT, CoAP, Md5.

1. INTRODUCTION

The Internet of Things (IoT) is a concept that aspires to extend the advantages of continuous internet connection for various things such as data sharing, remote control, and data monitoring. The Internet of Things (IoT) covers a wide range of technologies, from sensors, actuators, RFID, wireless sensor networks to the cloud storage. The motive of IoT is to connect the physical world and the information world through an interface between the user and the equipment. Communication protocols for acquisition of data that can be used are HTTP, Cap, XMPP and MQTT protocol. HTTP protocol must be used to handle the acquisition of data but it is used for exchanging the data therefore it is less suitable in IoT based applications. MQTT is used for IoT because it provides secure and high quality data. HTTP uses the most common message exchange pattern i-e request/response protocol while MQTT uses a publish/subscribe messaging protocol. In addition, MQTT receives the applications in the field of sensory networks, human communication and automotive communication hence it is more inferior than HTTP and Cap. However, MQTT specification recommends the use of TLS I-e Transport Layer Security to protect the transaction between publisher/subscriber. There are obvious limitations to this approach. TLS does not guarantee end-to-end encryption and is otherwise “too heavy” for resource-constrained devices. Perhaps, TLS adds

memory and energy overhead that cannot be support by constrained devices. Therefore, MQTT speciation proposed an AES and DES encryption algorithm for constrained device network.

1.1 HTTP

The Internet network is designed to communicate via HTTP (Hyper Text Transfer Protocol). A variety of information, from pictures to texts, is sent online daily. HTTP is like a basic protocol interface for transferring large amounts of data quickly, easily, and stable from server to user devices such as a browser. HTTP is built on TCP. HTTP ensures that data transferred from one device to another will not be compromised so that the authenticity of the transferred data is verified. HTTP is an open communication protocol that can be read by any devices designed for the HTTP protocol such as a browser or smartphone via a browser program. HTTP transactions consist of two parts: a request command (request) sent from client to server, and a response command (response) sent from server to client. The reply and request process is sent using a data block with a specific format known as HTTP Message.

1.2 MQTT

MQTT (Message Queuing Telemetry Transport) is a light weight messaging protocol that uses a public/subscribe messaging pattern. Sending MQTT request using IoT device if faster than sending request using HTTP as MQTT message can be as small as 2 bytes whereas HTTP uses headers which contains lots of information which other devices may not care about. Also, if you have multiple clients that are waiting for a request for HTTP, you will need to send a POST command to each client. While in MQTT, whenever a server gets information from one client, it will automatically distribute that information to each of the interested clients. In this paper we augment MQTT protocol with AES, DES and Md5.

2. LITERATURE SURVEY

According to the Research paper in 2016 titled 'Comparison with HTTP and MQTT on required network resources for IoT', In this paper author has discussed the difference between HTTP protocol and MQTT protocol. This paper compares HTTP performance with that of MQTT, a type of named based transfer protocol. Additionally, this paper suggests MQTT enhancements to work better.

According to the Research paper in 2019 titled 'Performance Comparison of IoT Communication Protocols', In this paper the main objective of author is to evaluate the performance of IoT communication protocols for the application layer: AMQP, CoAP, and MQTT.

According to the Research paper in 2019 titled 'A Review Paper on-MQTT Based Centric System for Energy Measurement, Monitoring and Control', In this paper the author grants an overview of Message Queuing Telemetry Transport (MQTT) protocol which is light weight.

According to the Research paper in 2019 titled 'Security, Privacy and Forensic Concern of MQTT Protocol', In this paper the author aims to give the overview of the security and privacy concerns of the Internet of Things by analyzing MQTT a light weight protocol.

3. METHODOLOGY

In this paper, The proposed system describes the use of MQTT protocol to transfer data in IoT devices. In IoT, data transmission between different devices is important because an IoT devices has to deliver an instruction to a further device to manage system. Compared to polling

protocol, Push protocol is the suitable message communication protocol for IoT devices as it works in less bandwidth network. HTTP, XMPP and Cap protocol were implemented through these push message services. But these push protocols require reliable network and is less effective in a situation where client sends request at irregular intervals of time.

MQTT client is a constrained node based on Arduino Uno. The Arduino Cryptography Library is used for encrypting, decrypting and authenticating MQTT PUBLISH message data. An MQTT client is able to serialize and de-serialize MQTT packets. Before publishing data, an MQTT-SN client firstly encrypts and generate authentication token based on AES and DES cryptographic functions. This packet is relayed to the MQTT Broker which is responsible for sending it to all subscribers. For MQTT client, the packet passes firstly to the MQTT Gateway. After the packet is de-encapsulated and serialized, it finally relayed to the MQTT client. The later decrypt the MQTT PUBLISH packet payload, then verify the authenticity of data.

While MQTT protocol can be used where network connection is unreliable and also it is useful in a situation where client sends request at irregular intervals of time. MQTT is an asynchronous protocol, therefore as soon as data is received in MQTT it can be send to the cloud.

3.1 Broker - Broker is a server that delivers information to interested clients connected to the server.

3.2 Client - A device that connects the broker to send or receive information.

3.3 Subject - The name to which the message is about. Clients publish, subscribe, or do both on the topic.

3.4 Publish - Clients send information to the seller to distribute to interested customers based on the topic name.

3.5 Subscribe - Clients tell the seller what topics they are interested in. When a client subscribes to a topic, any message published to the broker is distributed to subscribers of that topic. Customers can also unsubscribe to stop receiving messages from the broker about that topic.

4. DIAGRAM

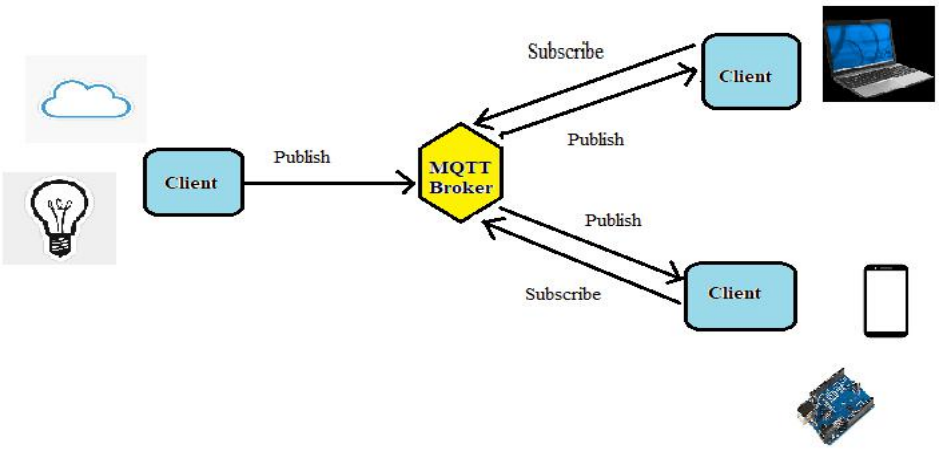

Fig 1: Architecture of MQTT protocol.

5. FUTURE SCOPE

5.1 MQTT can be used in large sensory network areas, to monitor earthquakes, floods, volcanic eruptions by transmitting a specific application sensor in a disaster prone area.

5.2 Instead of transferring data MQTT can also be extended to be part of a large network of energy monitoring systems.

5.3 MQTT can also be used in Laboratory monitoring system where the system monitors the vital condition.

6. CONCLUSION

The data transfer protocol used in IoT is defined in this context work. However, to use IoT more widely, lightweight communication protocols are required. Thus in this project, we purpose the use of Message Queue Telemetry Transport (MQTT) protocol to transfer data between devices (Arduino), As it is more secured and light weight protocol and provides greater efficiency.

7. REFERENCES

[1] OASIS MQTT Technical Committee, "MQTT Version 5.0," OASIS, Committee Specification 02, May 2018.

[2] H. Tschofenig and J. Arkko, "Report from the Smart Object Workshop," IETF, RFC 6574, Apr. 2012.

[3] A. Niruntasukrat, C. Issariyapat, P. Pongpaibool, K. Meesublak, P. Aiumsupucgul, and A. Panya, "Authorization mechanism for MQTT-based Internet of Things," in 2016 IEEE International Conference on Communications Workshops (ICC), 2016, pp. 290–295.

[4] S. Katsikeas et al., "Lightweight secure industrial IoT communications via the MQ telemetry transport protocol," in 2017 IEEE Symposium on Computers and Communications (ISCC), 2017, pp. 1193–1200.

[5] M. Ion, "Security of Publish/Subscribe Systems," Ph.D. dissertation, University of Trento, Italy, May 2013.

[6] B. S. Adiga, P. Balamuralidhar, M. A. Rajan, R. Shastry, and V. L. Shivraj, "An Identity Based Encryption Using Elliptic Curve Cryptog- raphy for Secure M2M Communication," in Proceedings of the First International Conference on Security of Internet of Things, ser. SecurIT '12. ACM, 2012, pp. 68–74.

[7] Ahlgren, B., Dannewitz, C., Imbrenda, C., Kutscher, D. and Ohlman, B., 2012. A survey of information-centric networking. IEEE Communications Magazine, 50(7).

[8] Amadeo, M., Campolo, C., Quevedo, J., Corujo, D., Molinaro, A., Iera,A.Aguiar, R.L. and Vasilakos, A.V., 2016. Information-centric networking for the internet of things: challenges and opportunities.

[9] Luzuriaga, J. E., Cano, J. C., Calafate, C., Manzoni, P., Perez, M., & Boronat, P. (2015, September). Handling mobility in IoT applications.