International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878, Volume-7 Issue-6S2, April 2019

Design of an Industrial IOT Architecture Based on MQTT Protocol for End Device to Cloud Communication

F.Jerald, M.Anand, N.Deepika

Abstract: Data transfer processes are extensively happened by HTTP. Nonetheless, this protocol reasons a big over head in networks for IOT. To solve this problem, internet of things protocols have been discussed. This paper discusses about the advantages of MQTT, machine-to-machine connectivity protocol, in IoT networks. Additionally, the paper proposes an improved IOT architecture using MQTT to achieve greater efficiency when connecting industrial networks with Internet based cloud plat forms.

Keywords: IOT, HTTP, MQTT, Blynk App, Cloud MQTT.

I. INTRODUCTION

Nowadays, various application sectors are mostly using Internet of Things (IoT). Interest in IOT is major concerned topic in all over the world. IoT comprises a huge quantity of small data blocks from the components like several type of sensors are transferred between networks. IOT would have few problems even though the Internet Protocol has been used for most type of communication. Currently, the application protocols over TCP/IP or UDP/IP needs Internet access. Hyper Text Transfer Protocol (HTTP), One of the application protocols standardized in IETF, [1-2], and General communication have been applied through Internet. However, when communication in IoT is applied by HTTP, a large quantity of tiny data blocks are transferred, resulting degradation in the performance are a serious issue. Moreover, IP addressing vary according to physical location, causing the problem of difficulty in network control. For solving these problems, following name-based architectures have been discussed [3 -9].Several challenges and opportunities related to the networking area focus on accepting this architecture to IoT [10]. MQ Telemetry Transport (MQTT) is a lightweight protocol for required network resources for IoT architecture and its performance is described in [11-13]. Some of the standard committees giving more attention to MQTT and related discussion were carried out on Name based routing and IP address based routing. From the survey of other research work the performance of MQTT with HTTP were studied. MQTT reduces protocol overheads and provides high efficiency communication for IoT. In this work the prospect of considering MQTT protocol as a way for allowing the communication process on the IoT platform.

Revised Manuscript Received on March 26, 2019

F.Jerald, Research Scholar, Dr.M.G.R Educational and Research Institute University, Chennai

M.Anand, Professor, Dr.M.G.R Educational and Research Institute University, Chennai

N.Deepika, Research Scholar, Dr.M.G.R Educational and Research Institute University, Chennai

II. HTTP FOR IOT COMMUNICATION

In IoT technology based communication HTTP is applied for transmitting large amount of tiny packets. Sometimes due to protocol overhead it causes troubles for example, increase in utilization of system resources and delay. The communication network configuration is given in figure 1.

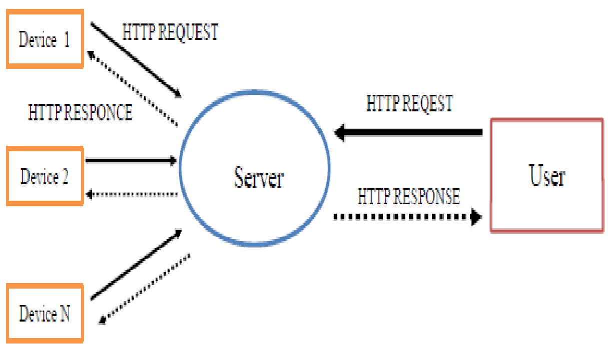

Fig. 1 System configuration using HTTP

The reliable communication is provided by HTTP, because it is functioned over TCP/IP. On the other hand the communication link is established by TCP are unrestricted for each access, while accessed data is transmitted which is derived from dynamic relationship of IP address and URL. The whole communication process is completed by means of persistent establishment of connection in the network. During the communication some problems causes due to overhead of protocol in IoT.

III. MQTT AND ITS PERFORMANCE

Protocol overheads similar to network resources and large delay in HTTP are reduced by MQTT. This part explained the operation of MQTT for IoT technology based communication.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

A. Summary of operations in MQTT

Transfer modes are classified into three types which are derived from essential dependability in MQTT. They are Non-assured transmission mode QoS1, Assured transmission mode and Assured service on applications. In perspective of reliability HTTP is related to QoS1at the same time it is a symmetric protocol. Since IoT technology is distributed in every device, it needs asymmetric communication based MQTT protocol for communicating with a server than HTTP [13, 15].

Publish and Subscribe are the two message sets present in MQTT. Publish message is used to send data packets on the other side it is received by subscribe message. Both the message sets are identified by topic in advance to transmit and receive the information. The MQTT system configuration is shown in Figure 2.

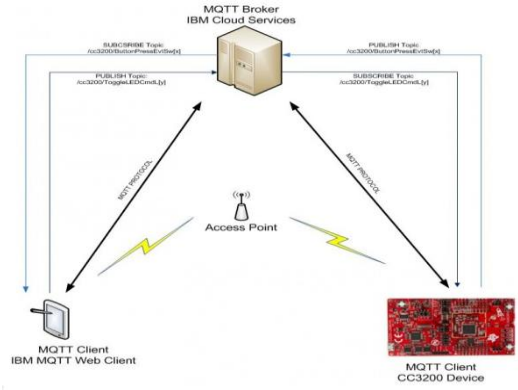

Fig. 2 System configuration using MQTT

In this configuration, communication sequence of devices by a user is shown in Figure 3.

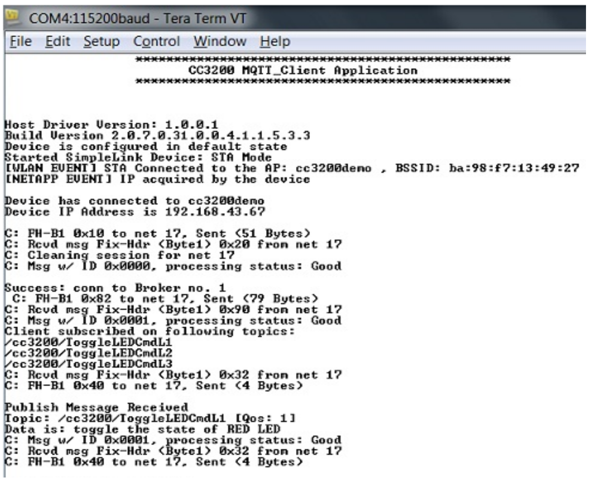

Fig. 3 Communication sequences of a device on MQTT

B. Advantages of MQTT

- Small code footprint
- Low network bandwidth requirement
- Faster response time
- Low power requirement
- Ease of scalability.

IV. CLOUD CONNECTIVITY USING MQTT

Cloud MQTT are managed Mosquitto servers in the cloud as shown in figure 4. Mosquitto implements the MQ Telemetry Transport protocol (MQTT), which presents lightweight methods of operation by means of a publish/subscribe message queueingre presentation.

In future every device in the world which is connected to the internet via machine-to-machine protocol reduces the design complexity. It is ideal for bandwidth sensitive applications, integrated systems and mobile phones.

Asynchronous communication is a full duplex communication occurred by message queues which will start to transmit a data from sender and received the message by the receiver at the same time. The transmitting messages are stored in the buffer, until the previous messages get received by the user. High-Quality Bandwidth sensitive applications are provided by MQTT and Mosquitto

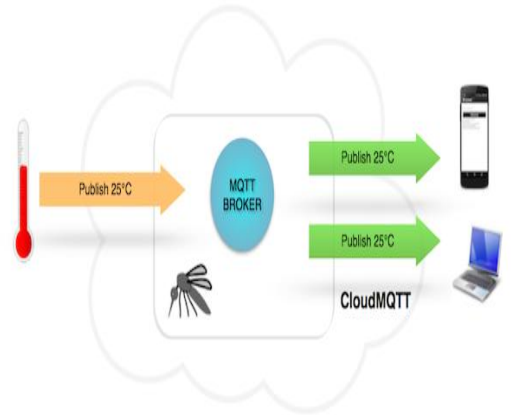

Fig. 4 Cloud connectivity using MQTT

V. BLYNK CLOUD DATA STORAGE AND MONITORING

Blynk user interface app was designed for the Internet of Things [14]. It can able to control hardware distantly from anywhere, it can display sensor data, and it can store data, visualize it and do many other things.

There are three major components in the platform:

1. Blynk App –It allows users to create interfaces for the projects using various widgets as shown in figure 5.

Fig. 5 Smart phone App based user interface

2. Blynk Server –It is significantly responsible for the communications between the smart phone and hardware.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

Everyone can use Blynk Cloud or run their private Blynk server locally. It's an open-source could easily handle thousands of devices and can even be launched on a CC3200 as shown in figure6.

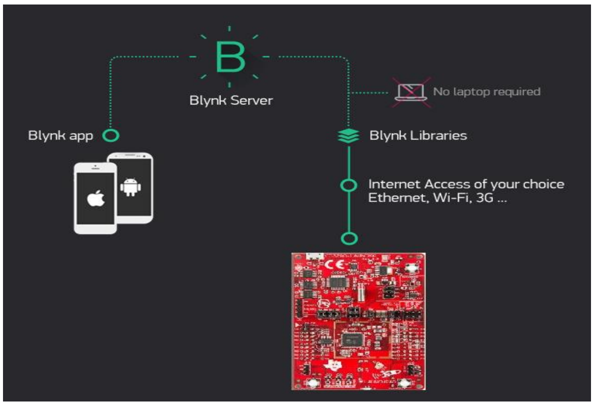

Fig. 6 Blynk server communication using CC3200 hardware

3. Blynk Libraries –It plays a major role for all the accepted hardware platforms - permit communication with the server and process all the incoming and out coming commands.

Blynk Protocol Message Format

Blynk server and our hardware device communicate using a custom TCP/IP protocol. Every message is consists of 2 parts.

Header:

1. Protocol command -1 byte;
2. MessageId - 2 bytes;
3. Body message length - 2 bytes;

Body: string (could be up to 2^15 bytes).

Blynk transfers binary messages through the subsequent structure:

Command field	Message Id field	Length/Status	Body
1 byte	2 bytes	2 bytes	Variable

a. Command field

Unsigned byte. Command field is 1 byte field which is responsible for storing command code from client, like login, ping, etc.

b. Message Id field

Unsigned short. Message Id field is a 2 bytes field for defining unique message identifier. It's used in order to distinguish how to manage responses from hardware on mobile client. Message ID field should be generated on client's side.

c. Length field

Unsigned short. Length field is a 2 bytes field for defining body length. It could be 0 if body is empty.

C. JSON format

JSON (JavaScript Object Notation) is a lightweight data-interchange format. It is easy for everyone to read and write.

It is easy for machines to parse and generate. JSON is a text format which is completely language independent but uses conventions that are familiar to programmers of the C-family of languages, including C, C++, C#, Java, JavaScript, Perl, Python`, and many others. JSON is an ideal data-interchange language based on the properties. JSON Client sends commands to the server and gets response for every command sent.

VI. CONCLUSION

The data transfer protocol used in IoT is explained in this work. IoT is expected to be applied to various applications asa social infrastructure. However, to deploy IoT widely, lightweight communication protocols are required. The paper concludes that MQTT protocol have advantages than HTTP in IoT network based on ICN architecture. Additionally, the paper proposed an improved IoT architecture using MQTT to achieve greater efficiency when connecting industrial networks with Internet based cloud platforms.

REFERENCES

1. Berners-Lee, T., Fielding, R. and Frystyk, H., 1996. Hypertext transfer protocol--HTTP/1.0 (No. RFC 1945).
2. Belshe, M., Thomson, M. and Peon, R., 2015. Hypertext transfer protocol version 2 (http/2).
3. Luo, J., Wu, C., Jiang, Y. and Tong, J., 2015. Name Label Switching Paradigm for Named Data Networking. IEEE Communications Letters, 19(3), pp.335-338.
4. Eum, S., Nakauchi, K., Shoji, Y., Nishinaga, N. and Murata, M., 2012. CATT: Cache aware target identification for ICN. IEEE Communications Magazine, 50(12).
5. Fang, C., Yu, F.R., Huang, T., Liu, J. and Liu, Y., 2015. A Survey of Green Information-Centric Networking: Research Issues and Challenges. IEEE Communications Surveys and Tutorials, 17(3), pp.1455-1472.
6. Ahlgren, B., Dannewitz, C., Imbrenda, C., Kutscher, D. and Ohlman, B., 2012. A survey of information-centric networking. IEEE Communications Magazine, 50(7).
7. Xylomenos, G., Ververidis, C.N., Siris, V.A., Fotiou, N., Tsilopoulos, C., Vasilakos, X., Katsaros, K.V. and Polyzos, G.C., 2014. A survey of information-centric networking research. IEEE Communications Surveys & Tutorials, 16(2), pp.1024-1049.
8. Yamamoto, M., 2015. Research trends on In-Network caching in content oriented networks. IEICE Technical Report., 115(461), pp.23-28.
9. Bari, M.F., Chowdhury, S.R., Ahmed, R., Boutaba, R. and Mathieu, B., 2012. A survey of naming and routing in information-centric networks. IEEE Communications Magazine, 50(12).
10. 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. IEEE Network, 30(2), pp.92-100.
11. Ahmed, S., Topalov, A. and Shakev, N., 2017, May. A robotized wireless sensor network based on MQTT cloud computing. In Electronics, Control, Measurement, Signals and their Application to Mechatronics (ECMSM), 2017 IEEE International Workshop of (pp. 1-6). IEEE.
12. Yokotani, T. and Sasaki, Y., 2016, September. Comparison with HTTP and MQTT on required network resources for IoT.In Control, Electronics, Renewable Energy and Communications (ICCEREC), 2016 International Conference on (pp. 1-6).IEEE.
13. http://www.blynk.cc/
14. Thangavel, D., Ma, X., Valera, A., Tan, H.X. and Tan, C.K.Y., 2014, April. Performance evaluation of MQTT and CoAP via a common middleware. In Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 2014 IEEE Ninth International Conference on (pp. 1-6). IEEE.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication