IARJSET

ISSN (Online) 2393-8021
ISSN (Print) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Vol. 8, Issue 7, July 2021

DOI: 10.17148/IARJSET.2021.8718

Terminal Chat Client using MQTT

Adithi S[1], Ayeesha Ruman[2], Mr Santhosh Kumar[3]

Student, Department of Electronics and Communication, KS Institute of Technology, Bangalore,India[1,2]

Assistant professor, Department of Electronics and Communication, KS Institute of Technology, Bangalore,India[3]

Abstract: In the era of rapid technological advancements, communication and information sharing has seen a major growth. Internet of Things (IoT) marks out a network of physical embedded objects (sensors, devices, networks, as well as different software technologies) that are used for exchanging data through different machines and networks on the Internet. Machine to machine communication has seen fast growth over the years and some protocols significantly in use are MQTT, AMQP and CoAP. Message Queuing Telemetry Transport (MQTT) is a lightweight protocol that supports a 'machine to machine' communication even with low bandwidths and high latency. MQTT follows a publish- subscribe model with a MQTT broker acting as the server used for communication. In this paper, we have used Paho MQTT client for publish and subscribe methods which is under the licensing of Eclipse Paho Project and Mosquitto MQTT broker -to design a terminal chat client that can establish communication between two laptops.

Keywords: MQTT, Chat client, IoT protocols, Paho MQTT Client, Mosquitto MQTT broker.

1.INTRODUCTION

Just like any emerging technology, the use of chat clients evolved rapidly in numerous fields including social media, customer support, Education, Health Care, Cultural Heritage, and banking sector. MQTT (message queuing telemetry transport) developed by IBM is a light weight messaging protocol that was designed for connecting power constrained devices over low bandwidth networks. Although MQTT was used as a proprietary protocol to communicate in oil and gas industry now it has a popularity in smart communication places and in the present, it is a leading open source protocol for connecting IoT. The latest version of the MQTT protocol specification, 3.11 became an OASIS standard. IoT has improved our lives as it enables us to collect monitored data from devices over a selected network without any human to human or human to computer interactions. MQTT can be addressed as an underlying communication protocol for IOT (Internet of Things). While we treat all embedded devices as a network in IoT, MQTT is a 'machine to machine' connectivity protocol. There are many real-world applications of IoT and this enables an entire sphere for improvement for any such application. A common example for an everyday use of a terminal chat window is that of Facebook which has its reach far across the globe. The basic concept of the MQTT protocol is "Subscribe or Publish or Both", on the specific topics (which can be considered as if choosing a channel of communication). For all the programming, we will be using Python programming language. The aim of this project is to develop an actively working Terminal chat client. For this purpose, we use MQTT protocols to design the chat window on the basis of two-way communication that includes both sending(publishing) and receiving(subscribing) the message, we use Mosquitto MQTT broker for the broker server. MQTT is an underlying communication protocol for IOT (Internet of Things). While we treat all embedded elements as a network in IOT, MQTT is a 'machine to machine' connectivity protocol. There are many real-world applications of IOT and this enables an entire sphere for improvement for any such application.

2.LITERATURE SURVEY

In this section, we summarise some of the important project related study that has been published in the past few years. A lot of research has been conducted on the working of MQTT protocol. In summary, the main components of MQTT architecture can be broken down into two parts- MQTT Client and MQTT broker. MQTT client can either be a publisher or subscriber and it establishes a network connection with the broker server. MQTT broker manages the distribution of messages and takes care of publishing, filtering, and storing the messages and details depending on the requirement. [1]

MQTT is a bidirectional communication protocol which helps in sharing data and also controlling devices, it also runs over the TCP/IP protocol and uses SSL/TSL connections for security management. MQTT offers different "Quality of Services" for varied requirements. [2]

MQTT protocol has three main entities- publisher, subscriber, broker. In one of the proposed security models for MQTT IoT protocol, user is considered to want to access data from the sensing device. The phases for this kind of framework would include the user registration phase, the device registration phase, and the user authentication phase. [3]

IARJSET
ISSN (Online) 2393-8021
ISSN (Print) 2394-1588
International Advanced Research Journal in Science, Engineering and Technology
Vol. 8, Issue 7, July 2021
DOI: 10.17148/IARJSET.2021.8718

For overcoming vulnerabilities further, an approach where message hashes that match an entry on a pre- generated table on the receiving node would be accepted. This can improve the accuracy with which better integrity can be achieved. [4]

3.SOFTWARE COMPONENTS

1.Eclipse paho

It is an MQTT python client library in which MQTT protocol versions 5.0, 3.1.1 and 3.1 can be implemented. This enables a client to connect to a MQTT to publish and subscribe messages.

2.Visual Studio Code

It is a source code editor developed by Microsoft for Linux, Windows and Mac OS which enables operations like task running, debugging and version control.

3.Mosquitto Broker

It is an open source message broker that is light weight and enables implementation on MQTT versions 5.0, 3.1.0, 3.1.1.

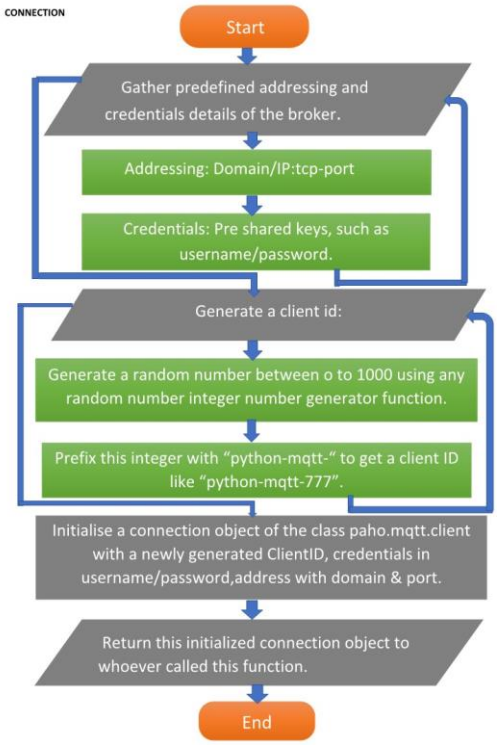
4.Python Programming

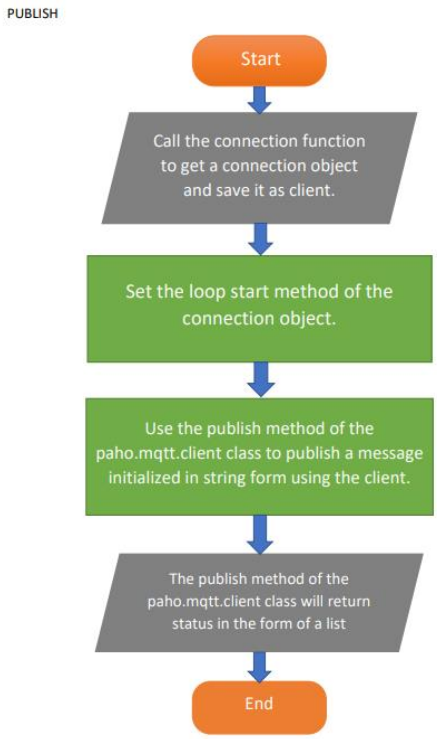
It is a widely used High level, Object Oriented Programming language which is simple and easy to learn. It also provides a wide range of libraries

4.WORKING

Connection: Gather predefined addressing and credentials details of the Broker.
Addressing: Domain/IP: tcp-port
Credentials: pre-shared keys, such as username/password
Generate a client id:

Here we generate a random number between 0 to 1000 using any random integer number generator function and prefix this integer with "python-mqtt-" to get a client ID like "python-mqtt-777". We initialise a connection object of the class paho.mqtt.client with a newly generated client ID, credentials in username/password, and address with domain & port and return this initialized connection object to whoever called this function.





IARJSET

ISSN (Online) 2393-8021
ISSN (Print) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

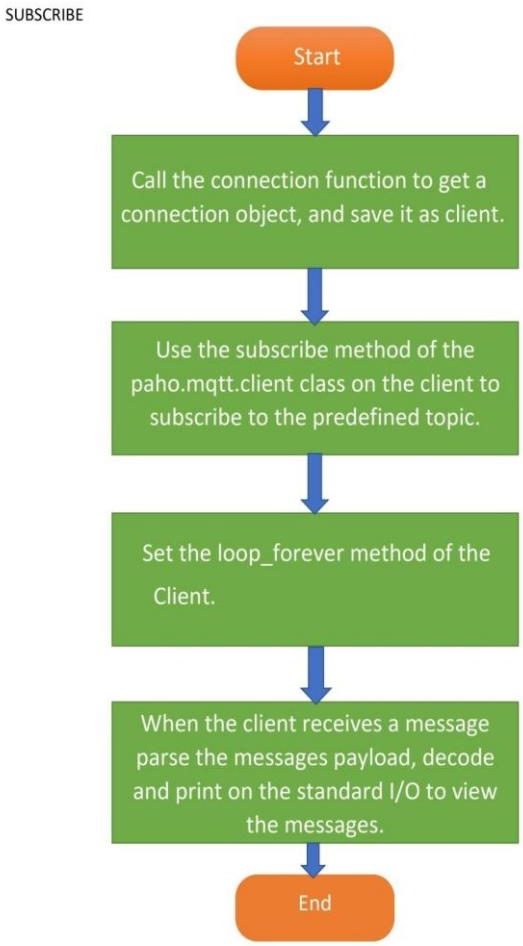
Vol. 8, Issue 7, July 2021

DOI: 10.17148/IARJSET.2021.8718

Publish: We call the connection function to get a connection object and save it as client and set the loop start method of the connection object. We use the publish method of the paho.mqtt.client class to publish a message initialized in string form using the client. The publish method of the paho.mqtt.client class will return status in the form of a list. The list is parsed to find values 0 or 1 for successful or failed attempt and the response message is printed accordingly.

Subscribe:

We call the connection function to get a connection object, and save it as client and use the subscribe method of the paho.mqtt.client class on the client to subscribe to the predefined topic and set the loop forever method of the client. When the client receives a message, the message payload is parsed, decoded and printed on the standard I/O device.



5.FUTURE ENHANCEMENT

To privatize the messages in case of text-to-text communication and reducing the risk of hacks present in IoT messaging through MQTT. Testing and trying to expand the extensibility of MQTT messaging servers. Building a good alternative interface of the server and implementation on raspberry pie.

6.RESULT AND CONCLUSION

In this paper we have discussed about the IoT in various industries and MQTT protocol is used for IOT applications. The main aim of this project is to make communication between the devices easy and privatized. We have experimented with different types of data that can be published between two different machines and understood the concept of master-slave machines and how communication works between these two and to came up with a protocol that lets the master publish a single message to all its slave machines we have also worked on the network security of the basic MQTT protocol and to used different libraries that allows implementation of IoT messaging that allows better communication and to develop an API based on TCP/IP protocol that also helps implementation on a different interface than a normal IoT environment.

7.REFERENCES

[1] Soni, Dipa & Makwana, Ashwin. (2017). A SURVEY ON MQTT: A PROTOCOL OF INTERNET OF THINGS(IOT).

[2] Mishra, Biswajeeban & Kertész, Attila. (2020). The Use of MQTT in M2M and IoT Systems: A Survey. IEEE Access. 8. 201071 - 201086. 10.1109/ACCESS.2020.3035849.

[3] Chintan Patel & Nishant Doshi. A novel MQTT security framework in Generic IoT model. Third International Conference on Computing and Network Communications (CoCoNet'19). Procedia Computer Science 171(2020) 1399-1408

[4] Dan Dinculeana and Xiaochun Cheng. Vulnerabilities and Limitations of MQTT Protocol Used between IoT Devices. Appl. Sci. 2019, 9, 848