MQTT -A LIGHTWEIGHT COMMUNICATION PROTOCOL RELATIVE STUDY

KALAIVANAN SUGUMAR[1]

[1]Affiliation not available

May 29, 2020

KEYWORDS

BROKER, INTERNET OF THINGS (IOT), MQTT, PROTOCOL

I. INTRODUCTION

In early 1999, during the transform of machine to machine communication MQTT was invented by Dr Andy Stanford-Clark of IBM, and Arlen Nipper of Arcom (now Eurotech) [1]. As the generation of computers evolved so did the demand for communication. Advances in the field of networking led to invention of MQTT.

MQTT (Message Queuing Telemetry Transport) is a communication protocol at the application layer. MQTT is widely used especially for IoT applications since it's a light weight protocol. It is claimed to be lightweight since it consumes very less bandwidth and battery loss is less. This makes it to be more suitable for IoT applications as IoT applications mostly involves with communication between small sensors and mini CPU like devices.

Unlike HTTP which works with request & response this MQTT works with publish-subscribe based messaging protocol. The publish-subscribe messaging pattern requires a message broker. The MQTT server is usually referred to as the broker. There are various types of MQTT broker like Mosquitto, HiveMqtt, Mosca, cloudMQTT, MQTT.Js.

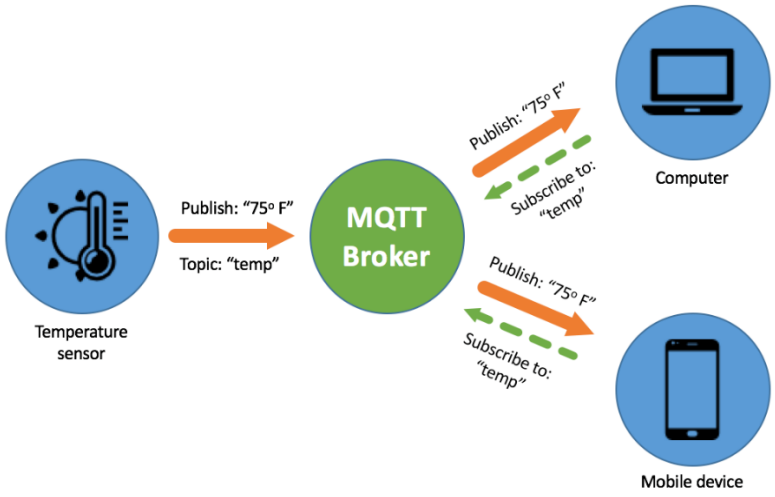

Fig.1 *MQTT Work flow representation*

The above Fig.1 represents how the MQTT broker works. The MQTT client connects to the MQTT broker with the subscribe command. All the information is organized in a hierarchy of topics. The client is subscribed to the particular topic name from where the messages have to be published. For instance, if user needs the temperature sensor value under the topic temp, then the client first subscribes to the topic temp. Then the broker fetches the value from the topic name and publishes it to the client.

MQTT relies on TCP/IP protocol for data transmission protocol. The client always interacts with the broker, but a system can have multiple broker servers that exchange data based on their current subscriber's topic.

II. CONCEPT OF MQTT AND BASIC PARAMETERS

BROKER

Like other communication protocols MQTT has a server which is referred to as Broker. The broker acts as a server which routes messages between client and device/sensor. There are many MQTT brokers available for testing and also for real time deployment.

TOPIC

MQTT topics are a form of addressing that allows MQTT clients to share information. MQTT topics are structured in a hierarchy similar to folders and files in a file system using the forward slash (/) as a delimiter. Users can create naming structures of their own choosing. There are certain limitations for choosing a topic name. Topic names should always be case sensitive, it should use UTF-8 strings and at least one character is to be valid. A client can subscribe to individual or multiple topics [2].

Example topic name structure:

home/hall/light

SUBSCRIBE

Unlike HTTP, MQTT does not work on request response basis. It works on publish subscribe method. Before sending a message to a device the client should have subscribed to a particular topic in the broker from where the data has to be fetched. It is not necessary to subscribe each and every time to a topic in the broker, once subscribed the message is received

PUBLISH

Once client has subscribed to the topic data gets published to the broker which in turn publishes it to client.

UNSUBSCRIBE

If the client does not require any data from a particular topic, then it can be unsubscribed from the broker.

III. SECURITY IN MQTT

For a secured data transmission MQTT provides authentication and TLS (Transport Layer Security). This TLS will create an encrypted connection between the MQTT client and MQTT broker similar to web browser client and web server. In this case we need a trusted server certificate on the client.

IV. COMPARISON

This section briefly explains the reason for choosing MQTT over HTTP and CoAP.

A) HTTP vs MQTT

HTTP(S) doesn't keep a connection open – so to get the "effect" of a continuous connection, the HTTPS client has to keep making poll requests, each of which involves creating the TCP connection and negotiating SSL/TLS encryption etc. each time a poll is needed. Whereas, MQTT makes a connection at the start, so

the TCP and SSL/TLS overhead cost is only paid once. MQTT has a keep alive message flowing between applications on top of the TCP connection, so that the applications can detect when the connection is broken, this allows the MQTT broker to reliably publish the (optional) client last will/testament when the connection is broken.

Stephen Nicholas compared MQTT vs HTTPS on Android 3G smart phones whose results are,

93x faster throughput

11.89x less battery to send

170.9x less battery to receive

1/2 as much power to keep connection open

8x less network overhead [3]

B) MQTT vs COAP

COAP is the Common Application Protocol which is similar to MQTT which works on publish subscribe basis. Constrained Application Protocol is a specialized Internet Application Protocol for constrained devices. It enables those constrained devices called "nodes" to communicate with the wider Internet using similar protocols. Comparison of MQTT and COAP was studied and presented in table 1.1.

MQTT
Consumes less battery
Consumes less bandwidth
Secured with TLS
Many to many communication
Communication only with subscribed topic
When client gets disconnected the server publishes the messages to the Will topic in the server side (LWT Last Will Testam
Supports publish subscribe basis

Table.1.1

V. MQTT PROS AND CONS

Each protocol has its own pros and cons so does MQTT.

A. Pros

Light weight API requires minimal processing on a device hence its very much suitable for devices that consumes low power. The message headers can be as small as 2 bytes which makes MQTT bandwidth efficient and ideal for spotty coverage or limited networks. MQTT-SNN (Sensor Networks) supports topic id instead of topic name and UDP, Zigbee, BLE and many other wireless protocols.

B. Cons

MQTT has no section for message properties and no support for such header fields as TTL. Besides it does not support message queuing.

VI. MQTT LOAD TEST RESULT AND DISCUSSION

MOSQUITTO LOAD TEST															
S.No	Load Test Name	Protocol	Host (Raspi)	Load Test Type	No.of messages to publish	No.of messages published	QoS Response	Published time (seconds)	No of instances	Topic	QoS	CPU Utilization(%)	Memory Utilization(%)	File size	Timeout (Seconds)	Runtime (Seconds)
1	Loadtest_eg1	MQTT	Mosquitto/1883	Publish	32	32	32	37	1	load_test_1	0	0.3	0.5	2 Bytes	60	10
2	Loadtest_eg2	MQTT	Mosquitto/1883	Publish	57	57	57	19	1	load_test_2	1	1.3	0.5			
3	Loadtest_1	MQTT	Mosquitto/1883	Publish	10	10		9.6	1	Sample	0	0.3	0.5	2 Bytes		
4	Loadtest_2	MQTT	Mosquitto/1883	Publish	100	100		55.3	1	Sample	0	0.3	0.5	2 Bytes		
5	Loadtest_3	MQTT	Mosquitto/1883	Publish	10	9		9	1	Sample	2	0.3	0.5	2 Bytes		
6	Loadtest_4	MQTT	Mosquitto/1883	Publish	10	10		9.17	1	Sample	0	0.3	0.5	45 Kb		
7	Loadtest_5	MQTT	Mosquitto/1883	Publish	100	100		54.7	1	Sample	0	0.7	0.5	45 Kb		
8	Loadtest_6	MQTT	Mosquitto/1883	Publish	100	100	100	10	1	Sample	2	1.7,2.3,2.6,3	0.5	45 Kb		
9	Loadtest_7	MQTT	Mosquitto/1883	Publish	10000	10000	4531	21	1	Sample	2	21.1	40.5	45 Kb		
10	Loadtest_9	MQTT	Mosquitto/1883	Publish	100	100	100	10.09	5	Sample	2	14	42	45 Kb		
11	Loadtest_10	MQTT	Mosquitto/1883	Publish	100	99	100	10.04	5	Sample	2	0.3	40	2 Bytes		
12	Loadtest_11	MQTT	Mosquitto/1883	Publish	100	99	100	9.9	1	Sample	2	0.7	40	5 kB		
13	Loadtest_12	MQTT	Mosquitto/1883	Publish	10000	10000	10000	15	1	Sample	2	11	45	5 kB		
14	Loadtest_13	MQTT	Mosquitto/1883	Publish	10000	10000	10000	16	1	Sample	2	5	50	5 kB		
15	Loadtest_14	MQTT	Mosquitto/1883	Publish	1 lakh	35496	35496	60	1	Sample	2	4.7	51	2 Bytes		
16	Loadtest_15	MQTT	Mosquitto/1883	Publish	10000	307	307	60	1	Sample	0	0.3	51	2 Bytes		
17	Loadtest_16	MQTT	Mosquitto/1883	Subscribe	10	NA	Timeout	NA	1	Sample2	0				Timeout	10
18	Loadtest_17	MQTT	Mosquitto/1883	Publish	1000	1000	1000	12	1	Sample	2	2	1.1	5 kB	60	10
19	Loadtest_18	MQTT	Mosquitto/1883	Publish	10000	10000	10000	29	1	Sample	2	15	7.2	5 kB	60	10
20	Loadtest_19	MQTT	Mosquitto/1883	Publish	1 lakh	35496	35496	60	1	Sample	2	17.8	23.9	5 KB	60	10

Table.1

The above table shows results of load test performed using MQTT. Following is the test bed that has been used.

Device: Raspberrypi 2

Broker: Mosquitto

Load test software: MqttBox

From table we infer that lesser is the QoS (Quality of Service) higher is the CPU utilization and higher is the QoS (QoS2) lesser is the CPU utilization. Irrespective of the number of messages. In the load test performed maximum of 1 lakh messages was to be published were each message is of size 2 Bytes out of which only 35,496 messages were published and rest of the messages were lost as the session timed out. The CPU consumption for 1 lakh messages was 4.7%. Similarly, 100 messages were to be published out of which all 100 was published were each message was of size 2 Bytes. The CPU utilization was 0.3%. Based on size of file CPU utilization tends to vary. Based on the analysis CPU utilization is less in MQTT when compared with other communication protocols.

VII. CONCLUSION

A brief discussion of MQTT is presented in this paper. A comparative study of MQTT with HTTP and COAP protocols was performed and is presented. Based on the load test performed this paper claims that memory and CPU utilization of MQTT is less when compared to other communication protocols.

VII. REFERENCES

[1] *MQTT* , 2019, MQTT.org, 20.01.2020, http://www.mqtt.org/news

[2] *Steves Internet Guide* , Steve Cope, 2011-2020, 20.01.2020 http://www.steves-internet-guide.com/mqtt/

[3] *Knoldus* , 2017-2020, 20.01.2020, https://blog.knoldus.com/iot-what-is-mqtt-how-it-is-lightweight/

[4] Hwang, H. C., Park, J., & Shon, J. G. (2016). *Design and implementation of a reliable message transmission system based on MQTT protocol in IoT* . Wireless Personal Communications, 91(4), 1765-1777.

[5] Dipa Soni., Ashwin Makwana., *A Survey on MQTT : A Protocol of Internet of Things* . International Conference On Telecommunication, Power Analysis And Computing Techniques. April 2017.

[6] Dan Dinculeana., Xiaochun Cheng., *Vulnerabilities and Limitations of MQTT Protocol Used between IoT Devices* , MDPI Applied Sciences, 27 February 2019.

[7] K.J. Reshmaa., J. Selvin Peter Paul., V. Swetha., *A Study on MQTT based Environmental Parameters, Monitoring and Alarming System,*Eurasian Journal of Analytical Chemistry, 06 December 2018

[8] P Gopi Krishna1, K Sreenivasa Ravi *IMPLEMENTATION OF MQTT PROTOCOL ON LOW RESOURCEDEMBEDDED NETWORK,* International Journal of Pure and Applied Mathematics, Volume 116

No. 6 2017, 161-166

[9] Sumit Pal., Sourav Ghosh., Sarasij Bhattacharya., *Study and implementation of environment monitoring system based on MQTT,*ENVIRONMENTAL AND EARTH SCIENCES RESEARCH JOURNAL Vol. 4, No. 1, March 2017, pp. 23-28