International Journal of Scientific and Research Publications, Volume 9, Issue 3, March 2019
ISSN 2250-3153

A Study on IoT Messaging Protocols and it's Comparision for implementation of IoT Services

Bhanujyothi H C, Rajesh S M, Vidya J and Sahana D S
Department of Computer Science and Engineering, GITAM School of Technology,
GITAM Deemed to be University, Bangalore Campus.

***Abstract*—Internet of Things (IoT) applications serves several areas, in which Real Time Applications has major siginificance and are widely used. In this paper, we discuss the implementation of IoT Services such as Room Temperature Monitoring and Fire Alarm System using Message Queue Telemetry Transportation (MQTT) broker. The MQTT Transportation broker has been utilized as a platform to provide the Internet of Things services. Open-source software component is used for connecting sensors and actuators to the platform by the IoT end device via Wi-Fi channel. We created smart Environment scenario and designed IoT messages satisfying the scenario requirement. A comparison of MQTT design and features with CoAP and AMQP protocols are discussed and tabulated. As a result MQTT stands ahead as effective protocol for IoT business applications in comparison with CoAP and AMQP.**

***Index Terms*—Internet of Things, MQTT, CoAP, AMQP, Smart Environment, Transportation Broker**

I. Introduction

INTERNET of things (IoT) is a novel model which is rapidly gaining the importance in business area and also in the modern wireless telecommunications with the integration of several technologies.This technology can be used in hospitals, homes, office, colleges and so on in order to control and report changes in the environment, which is affecting our daily lives significantly. Now a days IoT is acting like a daily need for objects over the Internet.

IoT is a worldwide system of physical and virtual things associated with the web. IoT encourages billions of devices, individuals and administrations to interconnect trade data and helpful information everywhere. Different communication and messaging protocols are used by IoT connected devices at different layers. While designing an IoT device, the selection of protocol largely depends on the type, layer and function to be performed by the device [6]. MQTT, CoAP, AMQP, XMPP and DDS are the few communication protocols used at the IoT application layer.

Networking with smart devices is increasing largely due to the ongoing technological revolution across the globe. People are using IoT and connected devices more and more to automate, control city traffic, industrial operations, track health, manage the fleet of vehicles, control home appliances, etc by using smart devices like smart phones, appliances, wearable devices, automobiles. Wearable devices use the internet to connect with other devices to exchange information with servers to perform different operations.

MQTT [1] is a publish/subscribe message exchange protocol developed by IBM. The MQTT system consists of MQTT broker and client. The MQTT broker is a message exchange platform that enables the message producer client to publish messages with a message identifier Topic. The MQTT broker delivers the topic messages, when the message consumer client subscribes to the Topic. Recently, MQTT has been adopted as the message transfer binding protocol in one of the M2M (machine to machine) IoT international standards.

Constrained Application Protocol (CoAP) [1] is an application layer protocol with a client-server architecture. This protocol runs over UDP (User Datagram Protocol). It is specifically developed for the resource-constrained devices. Clients and servers communicate through connectionless datagrams. It uses minimal resources and also useful in low power application. Derivation of SSL (Secure Socket Layer), DTLS (Datagram Transport Layer Security) protocol can be used for security of the messages.

Advanced Message Queuing Protocol (AMQP) [4] is used for sending transactional messages between servers, it is an open standard application layer protocol. As a message-driven middleware, it can process a large number of dependable queued exchanges. AMQP is mainly focused on efficient message delivery for end to end services. In this protocol messages can be transferred using TCP or UDP. AMQP centers around following messages and guaranteeing each message is conveyed as expected, regardless of failures or reboots.

II. Smart Environment and IoT Messaging Protocols

The functional capabilities of smart objects are further enhanced by interconnecting them with other objects using different wireless technologies [2]. Several research efforts have been conducted to integrate objects with IoT to form the smart environment. Implementation of a smart environment enhance the capabilities of smart objects by enabling the user to monitor the environment from remote sites based on different application requirements. The work on IoT-based smart environments can generally be classified into the following areas, in all these areas IoT play a very important role.

Figure 1 shows the IoT system for smart environment. IoT-based smart environments are: i) smart cities ii) smart grid iii) smart homes iv) smart building v) smart health vi) smart transport vii) smart industry.

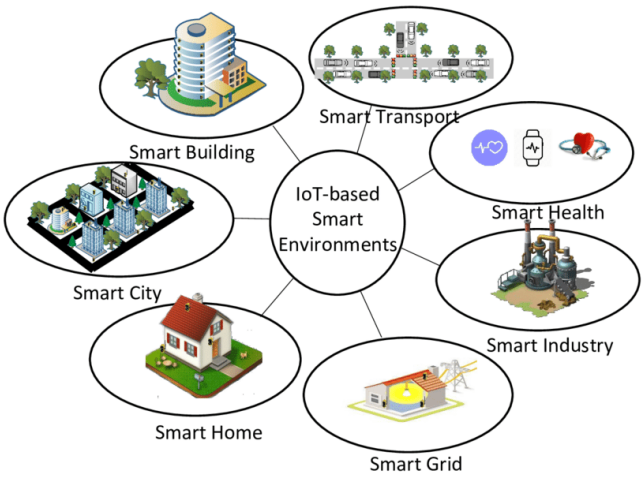

Fig. 1. Smart Environment's - using IoT System Design.

A. MQTT Protocol

MQTT is one of the commonly used protocols among IoT protocol at application layer. It is designed as a lightweight messaging protocol, that uses publish/subscribe operations to transfer data between clients and the server [3]. It consumes low power ,small size, minimized data packets and ease of implementation, these makes the protocol ideal of the machine-to-machine or Internet of Things world.

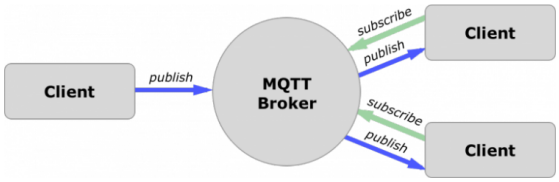

Fig. 2. MQTT protocol operation.

Figure 2 depicts MQTT protocol operation, like other internet protocol, it is also based on clients and a server. The server is responsible for handling the clients requests of receiving or sending data between each other. In MQTT protocol the clients are simply the connected devices and server is called as a broker, So when a device wants to send data to the broker, that operation is called as a *Publish*. When a device wants to receive data from the broker, that operation is called as a *subscribe*.

MQTT protocol specifically suitable for resource constrained networks. It is an efficient lightweight protocol, having ease of implementation in software and fast in data transmission. It sends minimized data packets results in low network usage and also it consumes low power hence we can say that MQTT is an energy efficient protocol. It is Flexible to choose Quality of Services with the given functionality and it is easy to implement. The main drawback of this protocol is Lack of encryption.

Several applications has been designed using MQTT protocol. Smart parking lot is an example application used to check the number and location of empty or vacant parking spots by installing parking sensors.

B. Constrained Application Protocol (CoAP)

CoAP has an interesting features specifically used for constrained devices. The CoAP protocol is a web transfer protocol which is used in resource constrained networks such as WSN, IoT, M2M etc, Hence the name Constrained Application Protocol [6]. The protocol is used for IoT devices having less memory and less power specifications. IoT is one of the most interesting and promising technology trends. It is an ecosystem where people, devices , objects are interconnected and exchange data. CoAP is a simple protocol with low overhead. This protocol is used in machine to machine (M2M) data exchange and is very similar to HTTP. CoAP protocol used in IoT mentions CoAP architecture, CoAP message header and message exchanges between CoAP client and CoAP server. CoAP Protocol exists between UDP layer and Application layer.

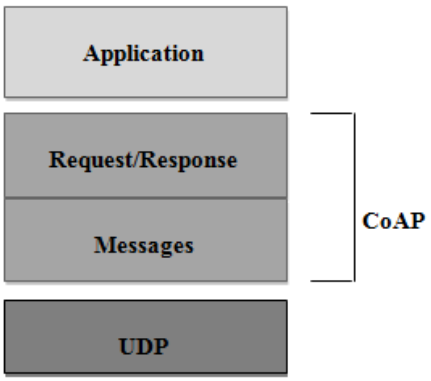

Fig. 3. CoAP Representation.

Figure 3 shows the representation of CoAP protocol [6], in this Request/Response and messages layers make CoAP protocol. The Request/Response layer manages request/response interaction based on request/response messages. The Messages layer handle with UDP and with asynchronous messages. CoAP supports four different message types:

1) Confirmable.
2) Non-confirmable.
3) Acknowledgment
4) Reset

CoAP Architecture extends normal HTTP clients to clients having resource constraints. These clients are known as CoAP clients. Proxy device bridges gap between constained environment and typical internet environment based on HTTP protocols. Same server takes care of both HTTP and CoAP protocol messages.

CoAP is a Web protocol used in M2M with constrained requirements so it supports URI and content-type. This is used for Asynchronous message exchange. This protocol is low overhead so very simple to parse. CoAP protocol is well designed protocol. It provides fast device-to-device communication as it sends small size of packet. It is a one-to-one

protocol, No broadcast message facility is provided. It is best suitable for Smart energy grids and smart homes applications.

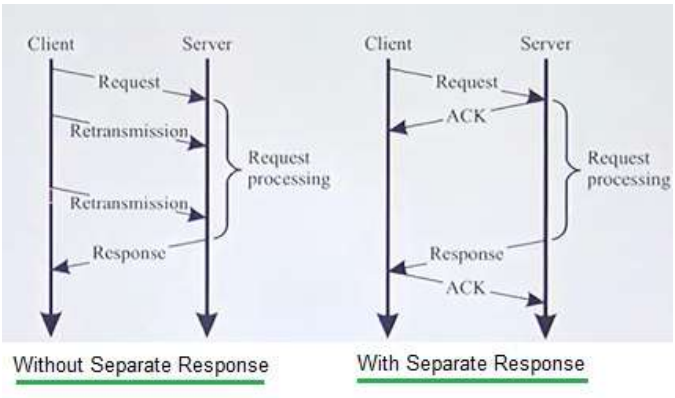

Fig. 4. CoAP Message exchange.

Figure 4 shows how message exchanged between client and server in CoAP protocol, There are two modes in which CoAP protocol messages get exchanged between CoAP client and CoAP server, they are without separate response and with separate response. With separate response, server gives notification to the client about receipt of the request message. This will help in avoiding unnecessary retransmissions but increase processing time. CoAP uses UDP so it is called as an unreliable protocol. Hence CoAP messages will get lost when they arrive at destination. To make CoAP as reliable protocol, stop and wait with exponential back off retransmission feature is incorporated in it. Duplicate detection is also introduced.

C. Advanced Message Queuing Protocol (AMQP)

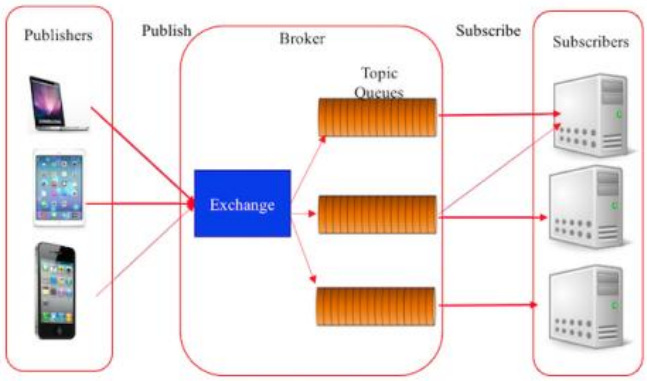

Fig. 5. AMQP Architecture.

AMQP runs over TCP layer, It is session layer protocol. Similar to MQTT protocol architecture, It is also based on publish/subscribe architecture. [7]

AMQP is different from MQTT based on their components, AMQP broker is composed of two components, namely exchange and queues, both are binded together.

Figure 5 describes AMQP client applications are known as producers (publishers) while AMQP server is known as broker. Client applications create messages which are given to broker. These client applications are known as consumers (subscribers). Messages are routed and are queued in the broker. These messages are being read by consumers from the queues where they are processed.

Exchange: This is a place where Publishers deliver messages. The messages contain routing keys which are used byexchange module in order to route them (i.e. messages). There are three different types of exchange methods, namely direct exchange, fanout exchange and topic exchange.

Queues: These are the places where messages are stored until they are delivered to or read by subscribers.

Binding: It states the connection between the message queue and the change.

AMQP can send messages over TCP and UDP layer. It provides an end-to-end encryption, so it is reliable. This protocol utilization relatively high resource i.e., power and memory usage, this is the main drawback of this protocol.

AMQP is mostly used in business messaging. It uses mobile handsets to communicate with back-office data centers. And it is one of the recently proposed protocols arising from the financial industry.

III. Smart Environment Design

A. Scenario

[1] In this section, we describe our smart environment IoT scenario for the reference implementation with MQTT in comparison with CoAP and AMQP protocols. In this environment, the cooling/heating unit (air conditioner) and the temperature sensor are interlocked so that the environment temperature can be controlled automatically. Fire detection sensor and sprinkler are installed to get fire alarm and to suppressing room temperature respectively. In the smart phone application, the environment temperature can be monitored remotely regardless of location. If a desired temperature is set on the application, the application sends the target value through IoT platform to the temperature control system. Then desired temperature is maintained by the local end device by controlling the air conditioner. Fire alarm, flame sensor, and sprinkler are also interlocked to provide automatic alarming, fire detection, and suppression service. IoT device automatically alarms the fire event when the flame sensor detects the fire and activates the sprinkler to suppress the fire. After suppressing the fire, it sends fire alarm message to smart phone application via IoT platform. The user of the application checks the fire suppression message and checks the operation status of the sprinkler and forcibly operates the sprinkler when not in operation.

B. Based on MQTT protocol

In this section we describe IoT message design based on MQTT, CoAP and AMQP protocol so as to suitable for the designing the above-mentioned smart Environment IoT scenario. [1]

The Figures 6 shows the IoT message exchange procedure according to MQTT protocol for room temperature controlling, fire sensing/alarming/suppression, and system status checking.

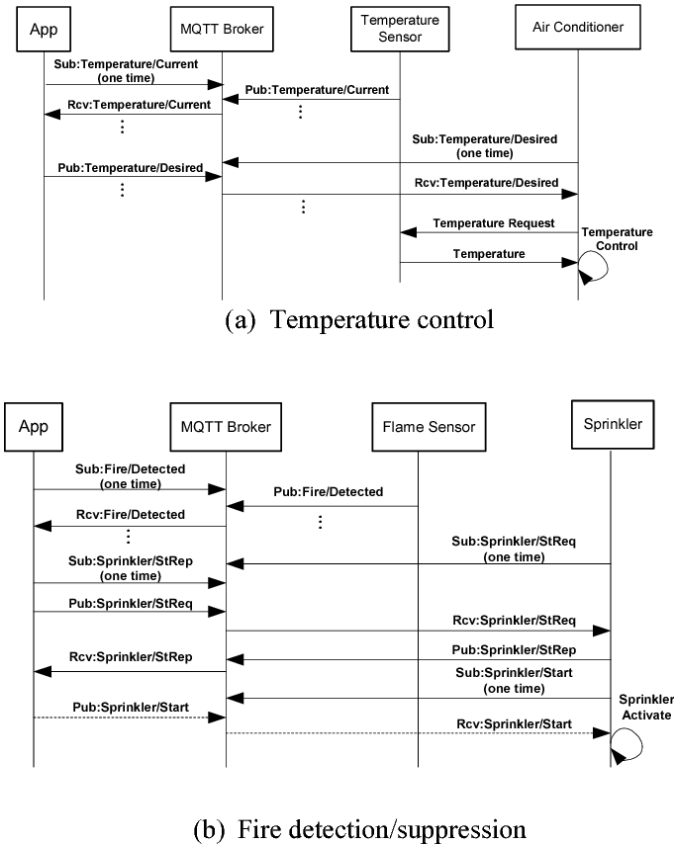

Fig. 6. IoT message exchange procedure.

The Figure 6 (a) shows the IoT message exchange procedure according to MQTT protocol for room temperature controlling, mobile application sends request to the temperature sensor via MQTT broker to know the current status of the temperature. Application is set with desired temperature, if the current environment temperature is not matching with the desired temperature then the application sends the target value to the temperature control system via MQTT broker. Then desired temperature is maintained by the air conditioner.

The Figure 6 (b) shows the IoT message exchange procedure according to MQTT protocol for fire sensing/ alarming/suppression. If temperature is not a targeted value set in the mobile application flame sensor detects the flame, when the flame sensor detects the fire, the IoT device automatically alarms the fire event and activates the sprinkler to suppress the fire, and sends fire alarm message to smart phone application via MQTT broker.

C. Based on CoAP protocol

The CoAP protocol uses two kinds of messages (i) Confirmable message (ii) Non-confirmable message. Figure 7 shows A CoAP confirmable message is a reliable message. When message exchanged between two end points, these messages can be reliable. Here a reliable message is obtained using a Confirmable message (CON). Confirmable message make sure that the client message will arrive at the server. A CoAP CON message is sent again and again until the other party sends an acknowledge message (ACK). The acknowledge message contains the same ID of the confirmable message (CON).

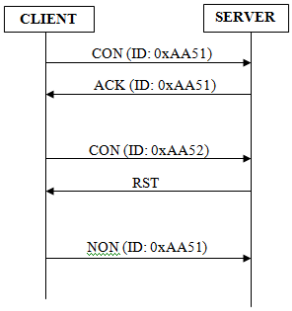

Fig. 7. CoAP Message Exchange.

In Figure 7, if server not receiving any request from the client, it can send back a Rest message (RST) instead of the Acknowledge message (ACK). Figure (c) Non-confirmable (NON) messages are other kind of messaging category. These kind of messages does not require an Acknowledge by the server. Here messages that does not contain critical information that must be delivered to the server and these messages are unreliable messages. Unreliable messages also have a unique ID.

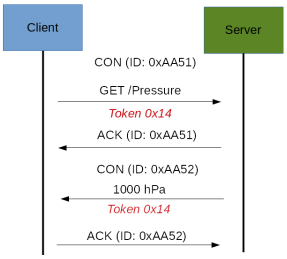

Fig. 8. CoAP Request/Response Model.

Figure 8 Depicts the Request/Response phase in CoAP protocol. The client is requesting the server and if the server can answer immediately to the client request then that request is carried using a Confirmable message (CON) then the server sends back the acknowledge message to the client that containing the response or the error code. In the CoAP message there is a Token. The Token is used to match the request and the response and it is different from the Message ID. [6]

If the server cannot respond to the request coming from the client immediately, then it sends an Acknowledge message with an empty response. When response is available then the server sends a new Confirmable message to the client containing the response. At that stage the client sends back an Acknowledge message.

IV. Comparison

A. Comparison between MQTT and CoAP

MQTT and CoAP are both useful as IoT protocols, but both have fundamental differences. MQTT is a many-to-many communication protocol for transmitting messages between

TABLE I
COMPARISION OF MQTT, COAP AND AMQP PROTOCOLS

Category	MQTT	CoAP	AMQP
Parameters	A lightweight Protocol easy to implement and fast in data transmission. Based on real time messaging technique. Minimized data packets. Low power consumption. So it saves the connected device energy.	Web protocol used in M2M with constrained requirements. Supports URI and content-type. Performs Asynchronous message exchange. Low overhead and very simple to parse.	Provides Infrastructure for a secure and trusted global transaction network. Well-stated message queuing and delivery semantics covering. Well-stated message ordering semantics describing. Supports Peer-to-Peer messaging across any network.
Advantages	Lightweight protocol and Flexible to choose Quality of Services with in the given functionality. Easy and quick to implement.	DTLS for security and Smaller packet size. Fast device-to-Device communication.	Messages can be sent over TCP and UDP. Provides end-to-end encryption.
Disadvantage	High power consumption due to the TCP-based connection. Lack of encryption.	As it is a one-to-one protocol there is no broadcast message facility . Reliability is applications responsibility.	Relatively high resource utilization i.e. power and memory usage. More power utilization as it uses TCP and UDP.
Application	Smart Parking System.	Smart energy grids and smart homes.	Mobile handsets, Communicating with back-office data centres.

multiple clients through a central broker. CoAP is a one- to-one protocol for passing state information between client and server. MQTT clients make a TCP connection to a broker. This usually presents no problem for devices. CoAP will send and receive UDP packets between clients and servers, devices may first initiate a connection to the head-end. MQTT protocol provides no support for labeling messages with types or other metadata to help clients understand it. Conversely CoAP provides inbuilt support for content negotiation and discovery allowing devices to probe each other to find ways of exchanging data. [6]

B. Comparison between MQTT and AMQP

AMQP protocol is a peer-to-peer protocol, this can be used between two peers and no need of broker in the middle. MQTT protocol is a lightweight protocol working only with a broker in the middle with no concept of queue. [4] AMQP protocol is more oriented to messaging than MQTT. [7]

V. CONCLUSIONS

We have discussed various IoT messaging protocols and implemented the same for smart environment scenario to evaluate the performance and efficiency. MQTT broker has been built and utilized as a smart environment IoT platform to build a room temperature control and fire alarm/suppression system. Here this system provides global access to IoT services and server maintenance difficulties can be eliminated. Here global access is possible without separately providing public IP, making it well suited for individual or small business. IoT service establishment based on the results of this study, we can conclude that MQTT are good technical candidates for small IoT business applications in comparison with CoAP and AMQP protocols. Future research may be subjected to the implementation of IoT services using DDS and XMPP.

REFERENCES

[1] D.-H. Kang, M.-S. Park, H.-S. Kim, D.-y. Kim, S.-H. Kim, H.-J. Son, and S.-G. Lee, "Room temperature control and fire alarm/suppression iot service using mqtt on aws," in *Platform Technology and Service (PlatCon), 2017 International Conference on*. IEEE, 2017, pp. 1–5.
[2] T. Malche and P. Maheshwary, "Internet of things (iot) for building smart home system," in *I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC), 2017 International Conference on*. IEEE, 2017, pp. 65–70.
[3] "Mqtt," https://1sheeld.com/mqtt-protocol/.
[4] 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," in *Consumer Communications and Networking Conference (CCNC), 2015 12th Annual IEEE*. IEEE, 2015, pp. 931–936.
[5] S. Pandikumar and R. Vetrivel, "Internet of things based architecture of web and smart home interface using gsm," in *International Journal of Innovative Research in Science, Engineering and Technology*, vol. 3, no. 3, 2014, pp. 1721–1727.
[6] D. Thangavel, X. Ma, A. Valera, H.-X. Tan, and C. K.-Y. Tan, "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*. IEEE, 2014, pp. 1–6.
[7] S. Vinoski, "Advanced message queuing protocol," *IEEE Internet Computing*, vol. 10, no. 6, 2006.

AUTHORS

First Author - Bhanujyothi H C, M.Tech, Assistant Professors, Department of Computer Science and Engineering, GITAM School of Technology, GITAM Deemed to be University,Bangalore Campus, India. e-mail: banu.banuchandrashekar@gmail.com

Second Author - Rajesh S M, [P.hD], M.E, Assistant Professors, Department of Computer Science and Engineering, GITAM School of Technology, GITAM Deemed to be University,Bangalore Campus, India. e-mail:rajeshsm.cse@gmail.com

Third Author - Vidya J, M. Tech, Assistant Professors, Department of Computer Science and Engineering, GITAM School of Technology, GITAM

Deemed to be University,Bangalore Campus, India. e-mail:vidya.sjprakash@gmail.com

Fourth Author - Sahana D S, M. Tech, Assistant Professors, Department of Computer Science and Engineering, GITAM School of Technology, GITAM Deemed to be University,Bangalore Campus, India. e-mail: ds.sahana16@gmail.com