Machine to Machine Communication Architectures Evaluation

B. V. S. Bharghava
Undergraduate Student,
Department of Computer Science and Engineering,
Kalasalingam Academy of Research and Education,
Krishnankoil, Tamilnadu, India.

Mrs. R. Kanniga Devi
Assistant Professor,
Department of Computer Science and Engineering,
Kalasalingam Academy of Research and Education,
Krishnankoil, Tamilnadu, India.

***Abstract* – This paper gives a basic ideology about the available architectures for developing Internet of things and its applications and helps to choose the best one as per the requirements. In order to develop an effective application, choosing the right architecture is very important. The current trend is moving towards the automation not only in one particular area, but in all other aspects like industrial automation IIOT, home automation and smart cities. So, in order to develop such environment, choosing the right architecture is very important. This paper discusses about some architectures like MQTT, XMPP, CoAP, DDS, mDNS, AMQP, OPC UA, which are application layer architectures.**

Keywords: Architecture, Evaluation, MQTT, XMPP, CoAP, DDS, AMQP, OPC UA, IIOT, Automation.

I. INTRODUCTION

As the world is moving towards digitization, there is a high demand of automation, making everything communicate with other things giving the scope for the rapid growth of the IOT. As the things keep on increasing, an effective architecture is very much needed. In the last two to three years everyone started talking that IPv6 will be replacing the existing networking address that is IPv4, but the current industry is still using the IPv4 for the device communication. But sooner or later the network will grow enormously, which will require large address spacing. So, there is a need that the architecture, which we choose will also support the IPv6 addressing.[1]The architecture should be platform independent and light weight as most of the IOT devices are battery operated. They need to effectively communicate using available resources. If the data transfer will take a lot of energy, then the application will not exist in the market for a long time. The main features required for the development of application follows:

- Application Architecture- The application architecture places a major role in the development of the application as it is the base for the development of the application [2].
- Communication- The application should be able to communicate with other devices to transfer the data.
- Lifetime- As most of the IOT devices are battery operated, it is needed that the life time of the device to be longer.
- Scalability- the Application should be scalable that is, even if there are large amount of devices added to the IOT, the current application should be capable enough to accommodate the new devices and should work even if the number of devices is reduced.
- Security- The security also plays very crucial role as all the devices in IOT are capable of transferring the data. They should ensure that data is being sent to right person.

II. IOT ARCHITECTURE

A. AMQP

The AMQP stands for Advance Message Queuing Protocol, which is an application protocol [17].
This architecture is much of a producer-consumer model, where the producer will keep on producing the data and there will be a broker in between who will be queuing the data to the consumers. Then the consumer will consume the data when required, it is depicted in Figure 1. The broker has mainly three functionalities namely.

- Exchange – The exchange uses the routing key present along with the message sent by the producer to route the message.
- Queues – This is the place where the data sent by the producer will be stored and the consumer will receive the message whenever he is willing to get the message.
- Binding – The binding help both the Exchange and the Queue by making a relation between them [20].

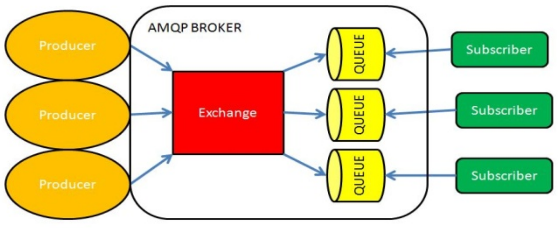


Figure 1 AMQP architecture

B. *OPC UA*

The OPC UA stands for OPC Unified Architecture is the communication protocol, which is developed by the OPC UA Foundation. The architecture is platform independent as in the Figure 2. The OPC UA has the following functionalities:

- Data Access – This allows the devices to communicate with other devices and collect the information gathered [3].
- Alarms and Events – This is much like an alert message which will be only sent under some events occurred.
- Data Exchange – This allows the devices to exchange the data which is stored in them.
- Historical Data Access – This allows the client devices to get the historical data that is stored in the server.
- Security – The OPC UA provides secure access to the data that is transferred over the network.

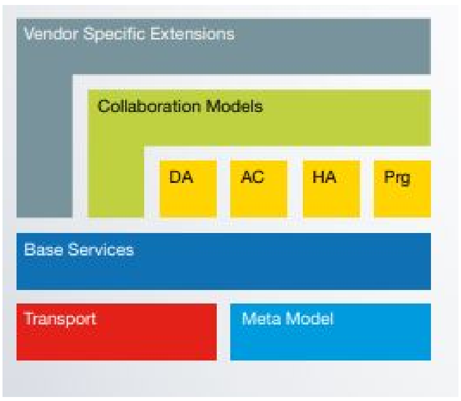


Figure 2 Layer of OPC UA

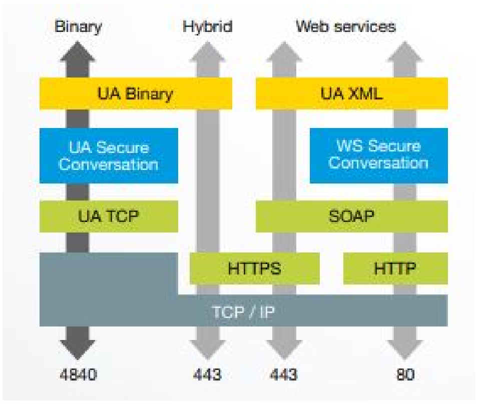


Figure 3 model of OPC UA

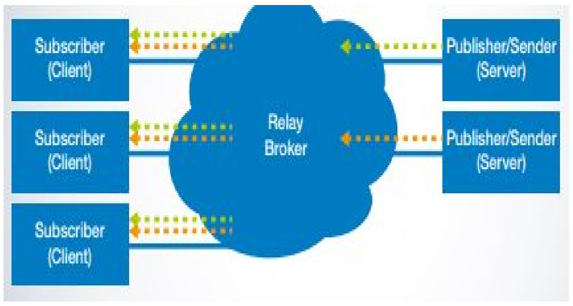


Figure 4 Pub Sub by OPC UA

The OPC UA uses the TCP communication [4], but it does the hand shake very quick and easily. The OPC UA also supports the Publisher subscription model Figure 4 for the session less transactions using the UDP packets for the transaction which will give a high latency rate compared to that of the normal client server model Figure 3. This architecture
is more unified so this architecture can be used for all the types of the IOT applications as it contain all the required features which are required for an application developed in the field of IOT.

C. *MQTT*

MQTT stands for message Query Telemetry transport. It is an ISO standard based protocol using publisher and subscriber [5]. MQTT is a light weight protocol. So, it is more suitable for the IOT. It requires a message broker for communication. MQTT works over the TCP/IP protocol. Machines can communicate using MQTT Figure 5.

MQTT communication involves four step connection, authentication, communication and termination. MQTT has two different ports 1883 and 8883. 1883 is an unencrypted message transporting port and 8883 in the encrypted message transporting port. The connection can be established by handshake method. During the process of hand shake, server certificate validation is done by the client for authenticating the server. Similarly, at the same time message broker will get the client certificate from the client to authenticate during messaging. As soon as the communication over, publisher/subscriber can send the disconnect message to terminate the communication causing graceful shutdown. If a subscriber/publisher had graceful shut down, then they can connect again easily.

C. 1 MQTT QoS

MQTT provides three different types of quality of service. At most once, exactly once, at least once [6]. Each of this quality of services provides different functionalities.

- Exactly once: Regardless of the message delivered or not, the message will be sent one time or none. This is also called unacknowledged messaging service. Where there will not be any communication between the publisher and broker as well as broker and subscriber.

- At least once: This is acknowledged messaging scheme. Where there will be guaranteed that the message has been delivered. For acknowledging purpose sequence numbers will be maintained. If there is no acknowledgement from the subscriber the message will be sent again. So, as to reduce the loss of messages acknowledged messaging scheme can be used. Duplication of messages can be formed in this type of quality of services.
- At most once: The other type of quality of service is at most once. The service is to prevent the supplication of the message. There is no need to monitor the messages. Because the message will not be sent again. Even though there is any loss.

C.2 MQTT architecture

MQTT architecture follows a publisher / subscriber model. As MQTT is used in IOT, the clients here are sensors or things. All these will be connected. Every sensor will act as client and the broker acts as server. Sensors will publish the messages in the topic. Topics are managed by brokers. The clients should subscribe to the topic to receive the messages published on topic. Clients may subscribed to many topics.

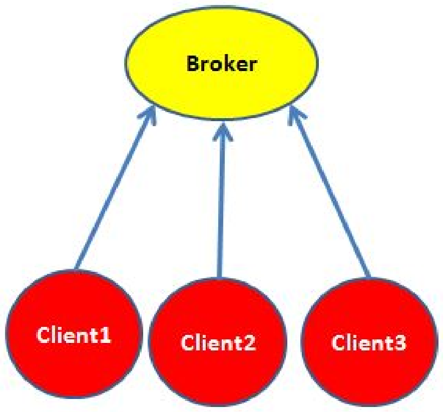


Figure 5 MQTT Architecture

D. XMPP

XMPP is an open source protocol used for voice or video calling and messaging. It is a spam free protocol. It is platform independent, secure, support instant messaging and compatible with other protocols [9]. XMPP considers itself as preferable over other protocols with the feature not only stick to IOT environment. It is a client server model Figure 6

XMPP follows a client server model, works similar to the some of the application layer protocols such as SMTP. The clients are connected to the server within a network. The messages will be transferred to the outside network using the gateway [11]. The Gateway is responsible for sending the messages to the outside network or between the domains. There is a chance of communication between the servers for routing. XMPP supports any type of protocol this is done by connecting all servers with XMPP gateway [10]. Through this gateway message will be transferred over the internet [12].

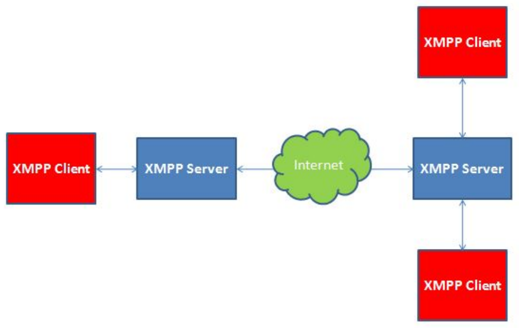


Figure 6 XMPP architecture

E. Constrained Application Protocol (CoAP)

CoAP an application layer protocol designed especially on IOT devices. It uses REST on the top of the HTTP. REST is a representational state transfer. The main applications using CoAP protocol are mobile and social network applications as in Figure 7.

CoAP protocol works in two different sub modules namely, messaging and request/response. The reliable communication can be provided by the messaging layer by detecting duplicate messages. On the other hand REST communication is handled by REST.

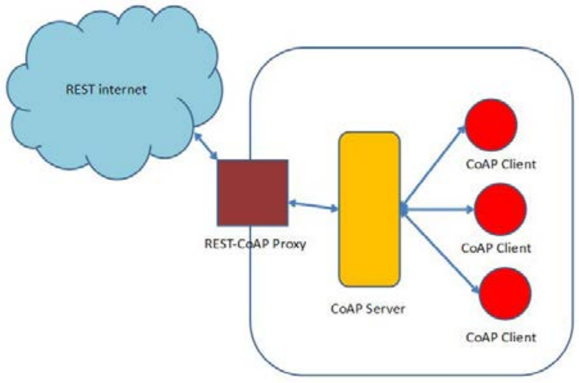
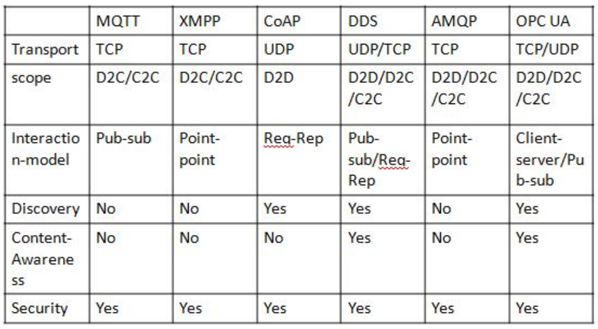


Figure 7 CoAP archirtecture

F. Data Distribution Service (DDS)

DDS a machine to machine communication protocol, which uses the publisher subscriber model [7]. This is more of a middleware that helps simplify the network programming. This communication protocol allows communication between the different language implementations. This protocol allows the users specify the Quality of service parameters configuring discovery. This protocol also switching between the primary and the secondary whenever there is a failure happened in the primary and switches back from secondary to primary when primary is recovered. This protocol supports many features like detecting who needs to receive the data where they are located [8]. But this protocol can any be used as a middleware which needs some other application running on top of it so this protocol cannot be used as an independent one.



	MQTT	XMPP	CoAP	DDS	AMQP	OPC UA
Transport	TCP	TCP	UDP	UDP/TCP	TCP	TCP/UDP
scope	D2C/C2C	D2C/C2C	D2D	D2D/D2C /C2C	D2D/D2C /C2C	D2D/D2C /C2C
Interactio n-model	Pub-sub	Point-point	Req-Rep	Pub-sub/Req-Rep	Point-point	Client-server/Pu b-sub
Discovery	No	No	Yes	Yes	No	Yes
Content-Awarene ss	No	No	No	Yes	No	Yes
Security	Yes	Yes	Yes	Yes	Yes	Yes

Figure 8 Comparison table

III. CONCLUSION

This paper discusses about different types of machine to machine communication protocol architectures [13]. Every architecture is important in its own way as you can see in Figure 8. It is not that only one architecture suites all the application needs depending on the application need we need to choose the architecture [14]. Like the AMQP architecture is more useful for the applications, where the servers keep on producing the resources and it is the client who will consume the resources whenever they are needed [16]. Next the MQTT architecture, which is very useful for a large enterprise, where there are many nodes connected and communication latency, is more important than the data reliability. The MQTT is capable of handling the large data transfers using the UDP packets where the producer will keep on generating the resources [15], but it is the subscriber who need to collect the data sent by the publisher and the data is not reliable as there is a high chance on loosing the data so this architecture best suites the non reliable communication with high latency [18]. The OPC UA architecture is more of a common architecture that is the name itself says that it is unified so this architecture will best suited for most of the application need but even in that we need to decide which is important to us if you want the data reliability then you can go for OPC UA client server or Discovery service architecture these both will give a reliable data communication [19]. If you want high latency but data reliability secondary then we can go for the OPC UA pub/sub architecture, which will work similar to that of the MQTT publisher subscriber model. So, before choosing the architecture which you are going to implement in your application understands the need of the application and then choose the architecture.

IV. REFERENCES

[1] “WSN Evolution in IoT”.
[2] “Application Architecture for IoT”.
[3] "IEC 61850 based communication OPC UA -The future smart grid automation."
[4] "OPC UA communication protocol"
[5] "A pub/sub protocol for WSN MQTT."
[6] "MQTT monitoring solution."
[7] "Introducing DDS.”
[8] "OPC UA handling data."
[9] "A compatible protocol XMPP."
[10] "Gateway using XMPP."
[11] "XMPP integration environment like home."
[12] "XMPP bridge internet."
[13] "M2M protocol architecture"
[14] "IoT architecture for all."
[15] "UDP high data transfer."
[16] "UDP packets over volte"
[17] "AMQP protocol for messaging."
[18] "Determinant latency of UDP."
[19] "OPC UA discovery service"
[20] "AMQP data binding"