A Survey: Embedded World Around MQTT Protocol for IoT Application

Kirit Vanani[1] Jignesh Patoliya[2] Hardik Patel[3]
[1]Reserch Scholar [2]Asst. Prof
[2]Department of Electronics Communication Engineering
[1]GTU PG School, Ahmedabad, India [2] CSPIT, Changa, India

Abstract— Now a day in a smarter embedded world, have IoT (Internet of Things). IoT have lot of things for the embedded systems, and it has the potential to transform our world with the help of it. IoT is playing role for smart cities, smart grid, smart health monitoring, smart clothes etc. All these technologies of IoT is going to change the living lifestyle of human being. MQTT (Message Queue Telemetry Transport) is a IoT standard protocol. It is Client-Server, based publish/subscribe messaging transport protocol. It is light weight, open source, simple, and can be designed such way that, we can easily implement in embedded devices, and IoT based systems/applications. These characteristics make it ideal for use in many situations, including IoT based various applications, embedded systems, and communication in Machine to Machine (M2M) where a small size code is required and/or network bandwidth is at appreciate.

Key words: IoT, MQTT, M2M, WSN, TCP, OASIS, ESB, CoAP, ETSI, ABE

I. INTRODUCTION

A. *Internet of Things*

The Internet of Things (IoT) is a cohesive part of the potential Internet and could be defined for productive global network infrastructure with self-configuring capabilities based on its standard and interoperable communication protocols where physical objects will be connected with Internet and identify themselves to other physical/virtual devices. In the IoT, physical/virtual devices are expected to become active participants in business, information exchange and social processes where they are enabled to interact and communicate among themselves and with the environment by exchanging data and information by sensed system environment. They responding autonomously to the "real/physical world" events and effecting it by running processes that trigger actions and create services with or without any direct human intervention/direction. [1]

More information about Iot is that, it is a smart network which connects all things to the internet for the purpose of exchanging information with agreed protocols. So, anyone can access anything, at any time and from anywhere. In IoT network, things or objects are wirelessly connects with smart tiny sensors.

IoT devices can interact with each other without or with human intervention. IoT have lot of things for the embedded systems, and it has the potential to transform our world with the help of it. IoT is playing role for smart cities, smart grid, smart health monitoring, smart clothes etc. All these technologies of IoT is going to change the living lifestyle of human being. [1]

B. *MQTT*

The MQTT protocol is lightweight in the sense that clients are small, and it uses network bandwidth efficiently. The MQTT protocol supports assured delivery and fire-and-forget transfers. In the protocol, message delivery is decoupled from the application. The extent of decoupling in an application depends on the way an MQTT client and MQTT server are written. Decoupled delivery frees up an application from any server connection, and from waiting for messages. The interaction model is like email, but optimized for application programming. [2]

MQTT is especially for machine-to-machine (M2M) and Internet of Things connectivity protocol. It was designed as a highly lightweight publish/subscribe messaging protocol and it is useful for connections with remote locations where a small size code is required and network bandwidth is at appreciate." [3]

MQTT is open source publish/subscribe protocol, for Wireless Sensor Networks (WSN). It is mainly designed in such a way that it can be run on competitive and battery-saving sensor/actuator devices and operate over bandwidth confinement WSNs. [4]

MQTT is publish/subscribe messaging protocol built on application layer stack of the Transmission Control Protocol (TCP) and designed to be lightweight. Moreover, it supports for the offline messaging to handle disconnected clients at online broker. It is in standardization within the Advancing Open Standard for the Information Society (OASIS) consortium. [5]

II. ARCHITECTURE OF IOT

The architecture of IoT consists of a set of components. All Layers of it, can be realized by means of specific technologies, and we will discuss for realizing each component. As shown in figure 1, there are also some upright layers such as devices manager, identity & access management.

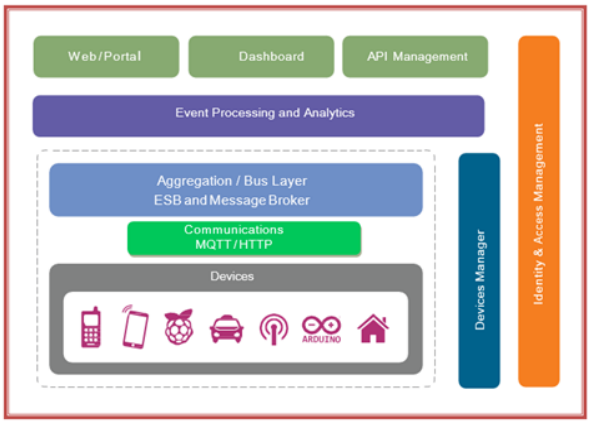

Fig. 1: Architecture for IoT

The layers are:

- Client (external communications) layers. i.e. Web/Portal, Dashboard, APIs
- Event processing and analytics layer (including data storage)

– Aggregation/bus layer – ESB (Enterprise Service Bus) and broker communication
– Relevant transports – MQTT/HTTP, etc protocols
– Devices (connected with IoT)

The upright layers are

– Device manager
– Identity & access management

A. Device Layer

The bottom layer of this architecture is the device layer. Devices can be of various types, but in order to be considered as IoT devices, they must have to communicate by indirectly or directly attaches to the Internet. Examples of this layer are:

– Arduino connected with Ethernet connection
– Raspberry Pi connected via Ethernet connection
– Smart car connected with wireless internet
– Android apps connected via a mobile phone

There are many more such examples of each type. [6]

B. Communication Layer

The communication layer supports the connectivity for the devices. There are multiple potential protocols for communication between the devices and the cloud. The most well- known three potential protocols are listed here,

– HTTP/HTTPS
– MQTT protocol
– Constrained application protocol (CoAP)

Here, for the reference architecture we have determine for select MQTT as the preferred device communication protocol, with HTTP as an alternative option. [6]

C. Aggregation/Bus Layer

This is important layer of IoT architecture. In it brokers communication play main role. This is an important layer for three reasons, which noted here:

1) The facility to support an HTTP server and/or an MQTT broker communicate with devices.
2) The facility for combine communications from different devices and route communications to a specific device.
3) The facility for bridge device and transform between different protocols, e.g. HTTP based APIs, which are settled into MQTT.

The aggregation/bus layer provides above capabilities as well as adapting into legacy protocols. [6]

D. Event Processing & Analytics Layer

This layer mainly takes the events from the bus and provides the ability to process and act upon these events. A core capability here is the requirement to store the data into a database. [6]

E. Clint/External Communications Layer

This layers includes three main approaches. Which are listed here:

1) We need the ability to create web-based front-ends and portals that interact with devices and with the event-processing layer.
2) We need the ability to create dashboards that offer views into analytics and event processing.
3) We need to be able to interact with systems outside of this network using machine to machine (M2M) communications using different APIs. [6]

F. Device Management

A device manager (server-side system) communicates with devices via various protocols and provides both individual and total control of devices. The device manager also needs to maintain the list of device identities and map these into owners. [6]

G. Identity & Access Management

The last layer is the identity & access management layer. This layer needs to provide the following services:

– Authentication taken of issuing and validation
– Other identity services like, OpenID Connect support for identifying inbound requests from the web layer
– Directory of users.
– Policy management for access controlling. [6]

III. MQTT Standard Protocol Of IoT

MQTT protocol used at session layer in IoT protocol stack. In order to utilize IoT paradigm, interconnected devices need to communicate using lightweight protocols which do not need extensive use of CPU resources. For this, C, Python, Java and MQTT scripting languages are preferable choices used by IoT applications. [7]

MQTT Broker controls the distribution of information. It stores, forwards, filters and prioritizes public requests from the publisher client to the subscriber clients. Clients switch between publisher role and subscriber roles depending on the functionalities desired. [7]

So many MQTT brokers available now, like, Apache ActiveMQ, Apache Apollo, IBM Message Sight, JoramMQ, Mosquitto, RabbitMQ, HiveMQ, MQTT.js, Web sphere MQ, Solace Message Routers, mosca, etc. All are different in their feature; they can be setting as per our requirement. Some of them we can also implement additional features. [8]

MQTT makes small attempt to enable device-to-device data transfer, also separate the data to many recipients (Clients). Since it has a clear, compelling single application, MQTT is simple, offering few control options. It also doesn't need to be particularly fast. In this context, "real time" is typically measured in seconds. [9]

The Publish/Subscribe model used in MQTT systems is very easily mapped to resource observers. Figure 2 shows a diagram of an MQTT Observer associated with a Smart Object Observable Property such that it can publish updates to the MQTT broker that result from PUT operations and subscribe to the MQTT broker updates and apply them to the Observable Property. [10]

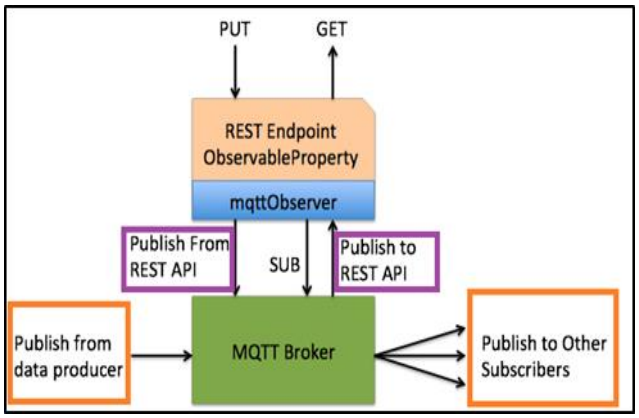

Fig. 2: MQTT Observer work in IoT

IV. Existing Approaches

Many researchers have been working on MQTT protocol for communication in embedded devices, machine to machine (M2M) communication and IoT framework where a small code is required and/or network bandwidth is at appreciate. In this section, we described various MQTT and IoT based system which are proposed in recent years.

A. MQTT - A Publish/Subscribe Protocol for Wireless Sensor Networks (WSN)

Hunkeler, U and team [4] proposed that, MQTT is designed in such a way that it can be run competitive and battery-saving sensor/actuator devices and operate over bandwidth constrict WSNs. A particular advantage of MQTT over other protocols is that it is based on a well-known publish/subscribe protocol already widely used now a day. MQTT allows a transparent data exchange between WSNs and any kind of networks and even between different WSNs. In addition, MQTT is extremely lightweight and can be further exposed down to a desert minimum.

B. IoT Application Layer Protocol Analysis over Error and Delay prone Links

Collina, M and team [5] research work is to understand in which conditions is better to use CoAP or MQTT in respect to the increase of delay and packet loss. They analyzed MQTT and CoAP protocols, both protocols suited for low-cost, low-power and resource controlled devices. As their results show, MQTT offers higher throughput and lower latency as compaire to CoAP in high offered traffic scenario, in the presence of high percentage of packet loss and delay. Based on their results, it is possible to assess a guideline in choosing the application protocol for an IoT application depending on the network characteristics. MQTT performs better in the low throughput scenario with a single device, they suggest for use of MQTT in presence of high delay and a single device.

C. Converging MQTT resources in ETSI standards based M2M platform

Hsiang Wen Chen and team [11] proposed a, method to integrate MQTT protocol with the ETSI (European Telecommunication Standard Institute) M2M architecture via a new network function called MQTT proxy. The MQTT proxy, on the one side, acts as an MQTT broker to the MQTT clients. While on the other side, it serves as a Gateway Application for interfacing with the ETSI M2M-compliant architecture. They also make comparison between the MQTT Proxy and the HTTP Proxy, it shows that the MQTT Proxy has lower latency, better power-saving and more support feature than the HTTP Proxy.

D. End-to-end Service Assurance in loT MQTT-SN

Govindan, K and team [12] inform that, MQTT for Sensor Network (MQTT-SN) can mainly have three different types of architectures.In Transparent Gateway architecture will have one Gateway for each sensor nodes. In Hybrid Gateway architecture, nodes can be connected to multiple Gatewaysand visa versa. In Aggregated Gateway architecture, there is one Gateway which aggregates all the nodes data and send them to the Server. They proposed that, MQTT-SN model the client is connected to Gateway using MQTT-SN over wireless UDP. The client can connect to multiple Gateways. The Gateways use conventional MQTT over wired TCP to publish/subscribe message to/from sever. The content can be delivered with 3 levels of QoS. The QoS-O is called At-most-once delivery, QoS-l is called Atlest once delivery and the third QoS-2 is called exactly-once delivery.

E. Secure MQTT for IoT

Singh Meena and team [13] proposed a Secure MQTT (SMQTT) which augments security feature for the existing MQTT protocol and its variants based on lightweight Attribute Based Encryption (ABE) over elliptic curves. The advantage of using ABE is because of its inherent design which supports broadcast encryption (with one encryption, message is delivered to multiple intended users) and thus suitable for IoT applications. ABE have two types:

1) Ciphertext Policy based ABE (CP-ABE) and
2) Key Policy based ABE (KPABE).

In general each of these schemes are different w.r.t the access policy, key management and are suitable for different kinds of applications. Thus these schemes for SMQTT from IoT perspective.

F. Handling Mobility in loT applications using the

MQTT protocol

Luzuriaga, J.E. and team [14] provide a solution to improve MQTT with an emphasis on mobile scenarios. They proposed enhancements to the MQTT protocol. Their proposal maintains the publish/subscribe approach but decouples the pure data generation process by the data sending process. It is based on a technique called intermediate buffering. This decoupling allows for recovery when the communication channel presents disruption periods, even if these are very frequent and last for several seconds, a situation where TCP fails to recover from. They prediction that, there is no information loss during the hand-off of the producer/publisher node, thus making a messaging system based on MQTT protocol robust, and able to guarantee message delivery without losses in the presence of publisher node mobility. Messages losses would be present only when roaming time tends to infinite, a situation prone to cause memory leaks and the system buffer capacity to be overloaded.

V. CONCLUSION

In this paper, we made an attempt to provide survey on the Embedded world and/or machine to machine (M2M) communication around MQTT protocol, for IoT application using various smart wireless sensors logger. As provide above existing approaches, we can get many information like,

- As Reference existing approach 1), we can use MQTT for WSN without anyking of technical problem.
- As Reference existing approach 2), we can use MQTT instead of CoAP protocol in IoT application.
- As Reference existing approach 3), we can use MQTT proxy instead of, HTTP proxy in IoT application.
- As Reference existing approach 4), we can use MQTT-SN for End-to-end Service Assurance in loT application.
- As Reference existing approach 5), we can get surety that, we can provide security in MQTT for IoT based application.
- As Reference existing approach 6), we can use MQTT for Large-Scale Cyber-Physical Systems in term of Adaptive Periodic Communication.
- As Reference existing approach 7), we can use MQTT protocol, for Handling Mobility in loT applications.

So, as per above survey, We can get surety that, for embedded world and/or Machine to Machine (M2M) communication we have to use MQTT protocol for various IoT application.

REFERENCES

[1] Internet of Things Europe, [Online]. Available: http://www.internet-of-things.eu/introduction (accessed on Dec 3, 2015).

[2] Andy Banks, and Dave Locke, "An introduction to the MQTT transport for Mobile Messaging & M2M" [Online]. Available:https://developer.ibm.com/messaging/2012/12/05/introduction-mqtt-transport-mobile-messaging-m2m/ (accessed on Dec 3, 2015).

[3] Lelylan Blog, "How to Build an High Availability MQTT Cluster for the Internet of Things" [Online]. Available: https://medium.com/@lelylan/how-to-build-an-high-availability-mqtt-cluster-for-the-internet-of-things-8011a06bd000#.sqh0fa4kk. (accessed on Dec 3, 2015).

[4] Hunkeler, U.; Hong Linh Truong; Stanford-Clark, "MQTT-S — A publish/subscribe protocol for Wireless Sensor Networks", A Communication Systems Software and Middleware and Workshops, 2008. Cited by: Papers (35) | Patents (1) IEEE Conference Publications COMSWARE 2008. 3rd International Conference on Year: 2008 Pages: 791 - 798, DOI: 10.1109/COMSWA-2008.4554519

[5] Collina, M.; Bartolucci, M.; Vanelli-Coralli, A.; Corazza, G.E. "Internet of Things application layer protocol analysis over error and delay prone links" Advanced Satellite Multimedia Systems Conference and the 13th Signal Processing for Space Communications Workshop (ASMS/SPSC) 2014, IEEE Conference Publications, Year: 2014 Pages: 398 - 404, DOI: 10.1109/ASMS-SPSC.2014.6934573

[6] Paul Fremantle, "A reference architecture for the internet of things." [Online]. Available: http://wso2.com/wso2-resources/wso2-whitepaper-a-reference-architecture-for-the-internet-of-things.pdf (accessed on Dec 3, 2015)

[7] Home of RF and Wireless Vendors and Resources, "What is MQTT in IoT" [Online]. Available: http://www.rfwireless-world.com/ Terminology/MQTT-protocol.html (accessed on Dec 3, 2015)

[8] MQTT Broker Feature Comparison, [Online]. Available: https://github.com/mqtt/mqtt.github.io/wiki/server-support. (accessed on Dec 3, 2015)

[9] Stan Schneider, "Understanding The Protocols Behind The Internet Of Things." [Online]. Available: http://electronicdesign.com-/iot/understanding-protocols-behind-internet-things#MQTT. (accessed on Dec 3, 2015)

[10] Michael Koster, "M2M Protocol Interoperability Using the Smart Object API" [Online]. Available: http://iot-datamodels.blogspot.in-/2013/10/m2m-protocol-interoperability-using.html1989. (accessed on Dec 3, 2015)

[11] Hsiang Wen Chen; Lin, F.J. "Converging MQTT Resources in ETSI Standards Based M2M Platform." Internet of Things (iThings), 2014 IEEE International Conference on, and Green Computing and Communications (GreenCom), IEEE and Cyber, Physical and Social Computing(CPSCom), IEEE Conference Publications Year: 2014 Pages: 292 - 295, DOI: 10.1109/iThings.2014.52

[12] Govindan, K.; Azad, A.P. "End-to-end service assurance in IoT MQTT-SN." Consumer Communications and Networking Conference (CCNC), 2015 12th Annual IEEE, IEEE Conference Publications Year: 2015 Pages: 290 - 296, DOI: 10.1109/-CCNC.2015.7157991

[13] Singh, Meena; Rajan, M.A.; Shivraj, V.L.; Balamuralidhar, P. "Secure MQTT for Internet of Things (IoT)." Communication Systems and Network Technologies (CSNT), 2015 Fifth International Conference. IEEE Conference Publications, Year: 2015 Pages: 746 - 751, DOI: 10.1109/CSNT.2015.16

[14] Luzuriaga, J.E.; Cano, J.C.; Calafate, C.; Manzoni, P.; Perez, M.; Boronat, P "Handling mobility in IoT applications using the MQTT protocol." Internet Technologies and Applications (ITA), 2015. IEEE Conference Publications, Year: 2015 Pages: 245 - 250, DOI: 10.1109/ITechA.2015.7317403

