A SURVEY OF PROTOCOLS USED IN IOT AT APPLICATION LAYER AND TRANSPORT LAYER

[1]Deepak Singh, [2]Harish Kumar
[1]M.Tech. Scholar, [2]Assistant Professor
[1] Department of Computer Engineering, JC Bose University of Science and Technology
YMCA, Faridabad, India

Abstract : Internet of Things is a most common and trending term between students and researchers. And it is a technology which could change the shape of world. Different types of devices required different type of protocol for communication. There are some problems in WSN like static allocation of sensor and nodes and gateway, small lifetime of participating nodes, loss of packets during delivery and end to end delay etc. To overcome these problems we require IOT based scenario. In IOT, TCP and UDP are used as transport layer protocols to achieve QoS for IOT devices. In IOT applications existing transport and security protocols like TCP/ UDP and TSL/DTLS are not suitable for connection overhead, latency, and connection migration. In this paper, we have highlighted the protocols that are used in the application layer and transport layer. We have discussed about the CoAP, MQTT, XMPP, LwM2M and AMQP.

IndexTerms – IoT, CoAP, MQTT, XMPP, LwM2M and AMQP Protocols.

1. INTRODUCTION

Internet of Things (IOT) is the most common and trending term and researcher do a lot of work in this felid. In IOT different type of network integration required for connecting wide range of devices and services. With the help of large number of sensors and actuators to the Internet we can make a connection between things and generate a scenario of smart things, such as smart home, smart grid, smart city, remote medical monitoring and industrial control etc. According to estimation by CompTIA, number of connected devices will surpass 50 billion by 2020 [1].Communication between IOT devices occurs with the help of IP and Transport layer protocols, in which it provide a protocol stack for inter and internet connectivity similar to Smartphone's and PC's . Internet Protocol (IP) have used for most type of communication, but it face problem with IOT , so it required set of protocols.

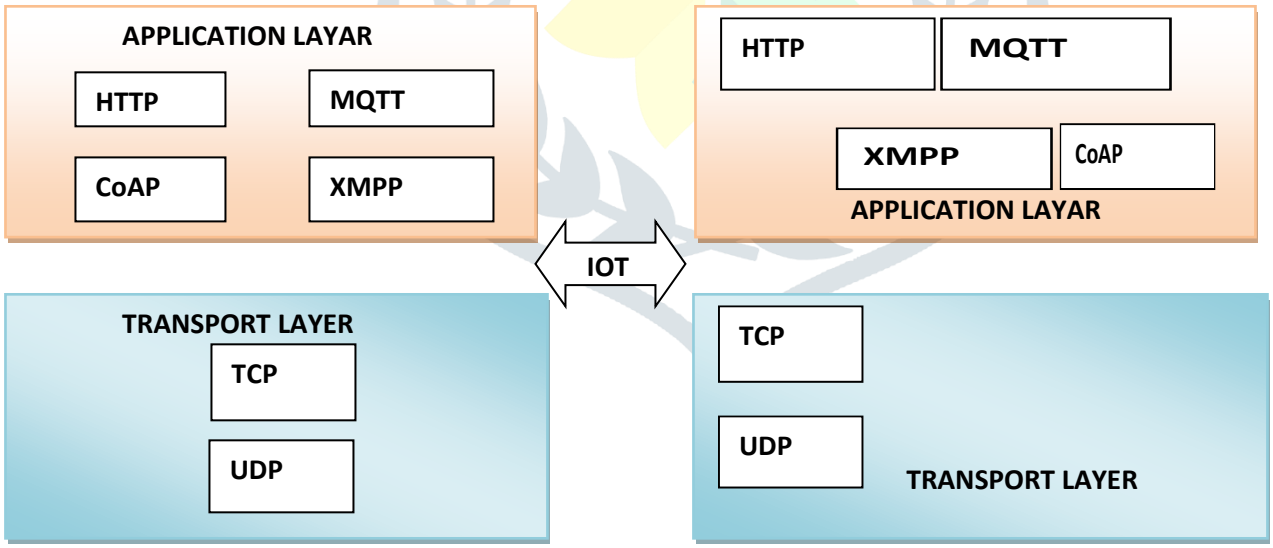

Figure 1 - IP-based Communication in IOT Model

For Application layer there are some protocols used for communication are CoAP (Constrained Application Protocol) ,MQTT (Message queuing telemetry transport), XMPP (Extensible Messaging and Presence Protocol) which use TCP provide reliable H2H connection oriented services and for transport layer ,there are TCP (Transmission Control Protocol) and UDP (User Datagram Protocol). Both TCP and UDP runs on yop of the internet protocol and are sometime referred to as UDP/IP or TCP/IP.Where UDP enables process-to-process communication, TCP supports host-to-host communication.

1.1 Things in the Internet of Things

In the basic structure of IoT there are things which connect to IoT Server via internet. There are various protocols and methods by which this connection is possible and each category have different kind of protocols for different kind of things for connection and communication.

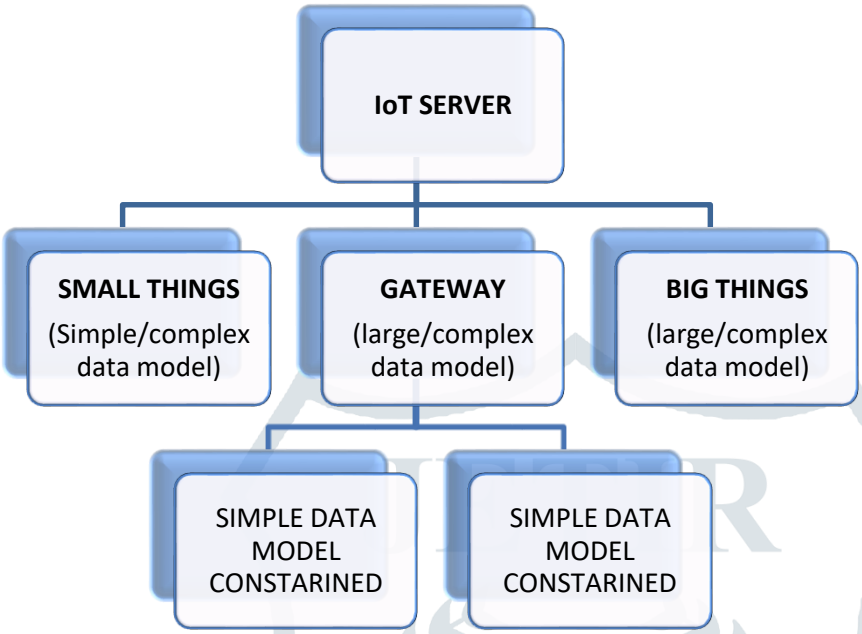

Figure 2 - Standard IoT Platform

1.1.1 Small Things

In this category there are smart things which are directly connected to the internet such as sensors, smart watch, smart locks and smart lights etc. Items in this category have limited memory and battery backup. They are connected to the internet with the help of SIM card and have a limited or expensive bandwidth. Small things have limited functionality.[2].

1.1.2 Small Things connected to the internet via Gateway

Devices which do not have "I" but referred as the IoT devices are those devices which directly not internet connected. Devices that communicate over the protocols like ZigBee , Z-Wave, LoRa , Sigfox and BLE these protocols consider as the IoT protocols . By Using a **Gateway** in the middle makes these things internet capable[2].

1.1.3 Big Things

Devices which have more complex design rather than size categorize as big things. In this the data model is more complex and need thousand parameters to managed. Modems , routers and industrial machines are in this category and have not restricted power supply and directly connected to Internet[2].

2. STANDARD IOT PROTOCOLS

2.1 COAP

The Internet Engineering Task Force (IETF) designed Constrained Application Protocol (CoAP) [3]which is a synchronous request/response application layer protocol runs over UDP.udp application layer protocol to manage resources by removing TCP overhead and reduce bandwidth requirement .It provide resource-oriented interactions in client server architecture by using HTTP commands DELETE, GET ,POST and PUT.

In CoAP to achieve reliability it integrated its own mechanisms because UDP is unreliable .QoS level achieve by adding two bits in header of each packet .

There are 4 type of message :-

2.1.1. **Confirmable**: A request message that requires an acknowledgement (ACK). The response can be sent either synchronously (within the ACK) or if it needs more computational time, it can be sent asynchronously with a separate message.

2.1.2. **Non-Confirmable**: A message that does not need to be acknowledged.

2.1.3. **Acknowledgment**: It confirms the reception of a confirmable message.

2.1.4. **Reset**: It confirms the reception of a message that could not be processed.

There is also a simple Stop-and-Wait retransmission mechanism for confirmable messages and a 16-bit header field in each CoAP packet called Message ID which is unique and used for detecting duplicates.

CoAP is created for IOT and M2M communication but it doesn't contain any security features.

DTLS(Datagram Transport Layer Security) protocol proposed to secure CoAP transactions .It provides authentication, data integrity, confidentiality, automatic key management, and cryptographic algorithms [4]. Even though DTLS secures UDP transfers, it was not designed for the IoT, thus its suitability can be argued. To begin with, DTLS does not support multicast [4], which is a prime advantage of CoAP compared to other application layer protocols. DTLS handshakes [5] require additional packets that increase the network traffic, occupy additional computational resources, and shorten the lifespan of mobile devices that run on batteries, an essential part of the IoT. CoAP is HTTP compatible when its designed for IoT but it create confusion when CoAP over DTLS. Other protocols (IPsec, Lithe) for securing CoAP can be found in the literature including approaches that are still being under research [4]-[5].

2.2 Message Queue Telemetry Transport (MQTT)

Message Queue Telemetry Transport (MQTT) protocol is an application layer protocol designed for resource-constrained devices [4]. MQTT is a messaging protocol and it is very fast and light. It uses a topic-based publish-subscribe architecture that runs on top of the TCP stack. Publish/subscribe protocols meet better the IoT requirements than request/response MQTT is designed to have a lower protocol overhead[6]. In MQTT there is a broker (server)[7] A_Survey_on_Application_Layer_Protocols_for_the_Internet_of_Things_Transaction_on_IoT_and_Cloud_Computing.pdf] that contains topic. MQTT is a "pub-sub" protocol that works similarly to WhatsApp or chat applications. In this protocol client publishes a message B on topic C then all the clients who subscribe for the topic C will receive the message B extremely quickly.It has a lower protocol overhead as compare to HTTPand TCP-based application layer protocol. The reliability of messages in MQTT is taken care by three Quality of Service (QoS) levels:

2.2.1. **QoS0** (Fire and forget): A message is sent once and no acknowledgement is required[8].

2.2.2**. QoS1 (**Delivered at least once): A message is sent at least once and an acknowledgement is required [8].

2.2.3. **QoS2** (Delivered exactly once): A four-way handshake mechanism is used to ensure the message is delivered exactly one time [8].

MQTT brokers may require username/password authentication which is handled by TSL/SSL which ensure security in IoT communication.

2.3 Lightweight M2M (LwM2M):

Created by the Open Mobile Alliance, it is a fast, light and structured, session-based protocol. LwM2M enables most IoT functionalities while maintaining Device Management capabilities on restricted devices[2].

2.4 XMPP :

The Extensible Messaging and Presence Protocol (XMPP) was designed for messaging and chatting . It enables the near-real-time exchange of structured yet extensible data between any two or more network entities. For the new arising data applications it is not sufficient to provide required services. For this reason, Google has been stopped supporting the XMPP standard due to the lack of

worldwide support [9]. However, XMPP has re-gained a lot of attention as a communication protocol suitable for the IoT. XMPP runs over TCP and provides request/response and publish/subscribe messaging systems. It supports low latency message exchange and small message footprint [10]. Security provided by XMPP is TLS/SSL that is built in its specification. However, it does not provide QoS options that make it impractical for M2M communications. In XMPP the inherited mechanisms of TCP ensure reliability.

2.5 AMQP:

The Advanced Message Queuing Protocol (AMQP) is a protocol that can utilize different transport protocols. AMQP is reliable as TCP protocol [11]. AMQP provides non parallel publish/subscribe communication with messaging. Even after network disruptions AMQP provide store and forward characteristic which ensure reliability [12].

With the following message delivery guarantees its ensures reliability [11]:

2.5.1. At most once- a message is sent once either if it is delivered or not.

2.5.2. At least once- a message will be definitely delivered one time, possibly more.

2.5.3. Exactly once- a message will be delivered only one time.

Security is handled with the use of the TLS/SSL protocols over TCP.Recent research has shown that AMQP has low success rate at low bandwidths, but it increases as bandwidth increases [12]. Another study shows that comparing AMQP with the aforementioned REST, AMQP can send a larger amount of messages per second [13]

3 CONCLUSIONS

In this paper presented a approach for the IoT architecture where we describing the need and use of the transport layer and application layer protocol role and functionality.

Protocol	**Transport**	**QoS options**	**Architecture**	**Security**
CoAP	UDP	YES	Request/Response	DTLS
MQTT	TCP	YES	Publish/Subscribe	TLS/ SSL
XMPP	TCP	NO	Request/Response Publish/Subscribe	TLS/ SSL
AMQP	TCP	YES	Publish/Subscribe	TLS/ SSL

In this we conclusion that these light weight protocols are used for fast communication and the performance of IoT scenario can be enhanced. But each protocol have some drawbacks we identified that CoAP that runs over UDP is the most light weight protocol but in this there is no guaranty of successful delivery of packets. On the other hand MQTT , XMPP and AMQP are public/subscribe protocols and there are more reliable. MQTT is more energy efficient and more appropriate for devices which are battery operated. The computational and communication ability and energy consumption and security should be taken in consideration when choosing the most appropriate protocol, for this there is a possibility of creating a space on cloud that supports multiple protocol and select protocol according to the condition of the network and the device we used in the communication.

REFERENCES

[1] I.CompTIA, "Sizing up the internet of things", (2015) [EB/OL], https://www.comptia.org/resources/sizing-up-the-internet-of-things.

[2] Understanding IoT Protocols, Clients, and Servers. November 2017 Ohad Oman, VP Product & Business Development Friendly Technologies.

[3] Maria Rita Palattella, Nicola Accettura, Xavier Vilajosana, Thomas Watteyne, Luigi Alfredo Grieco, Gennaro Boggia, Mischa Dohler, "Standardized Protocol Stack for the Internet of (Important) Things", Communications Surveys & Tutorials IEEE (Volume:15 , Issue: 3), 2013, pp. 1389 – 1406.

[4] Thamer A. Alghamdi, Aboubaker Lasebae, Mahdi Aiash, "Security Analysis of the Constrained Application Protocol in the Internet of Things", Second International Conference on Future Generation Communication Technology (FGCT), 12-14 Nov. 2013, pp. 163 – 168.

[5] Shahid Raza, Hossein Shafagh, Kasun Hewage, René Hummen, Thiemo Voigt, "Lithe: Lightweight Secure CoAP for the Internet of Things", Sensors Journal, IEEE (Volume: 13, Issue: 10), Oct. 2013, pp. 3711 – 3720.

[6] http://mqtt.org/2011/08/mqtt-used-by-facebookmessenger, cited 28 Jul 2014.

[7] Shinho Lee, Hyeonwoo Kim, Dong-kweon Hong, Hongtaek Ju, "Correlation Analysis of MQTT Loss and Delay According to QoS Level", International Conference on Information Networking (ICOIN), 28-30 Jan. 2013, pp. 714 – 717.

[8] Vasileios Karagiannis1, Periklis Chatzimisios1, Francisco Vazquez-Gallego2, Jesus Alonso-Zarate2 " A Survey on Application Layer Protocols for the Internet of Things ", 1 CSSN Research Lab, Department of Informatics, Alexander TEI of Thessaloniki, Greece basilkaragiannis@gmail.com, peris@it.teithe.gr 2 Centre Tecnologic de Telecomunicacions de Catalunya (CTTC), Spain [francisco.vazquez, jesus.alonso]@cttc.es

[9] http://www.zdnet.com/google-moves-awayfrom-the-xmpp-open-messaging-standard7000015918/, cited 28 Jul 2014.

[10] Sven Bendel, Thomas Springer, Daniel Schuster, Alexander Schill, Ralf Ackermann, Michael Ameling, "A Service Infrastructure for the Internet of Things based on XMPP", IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops), 18-22 March 2013, pp. 385 – 388.

[11] http://en.wikipedia.org/wiki/Advanced_Message _Queuing_Protocol, cited 28 Jul 2014.

[12] Frank T. Johnsen, Trude H. Bloebaum, Morten Avlesen, Skage Spjelkavik, Bjørn Vik, "Evaluation of Transport Protocols for Web Services", Military Communications and Information Systems Conference (MCC), 7-9 Oct. 2013, pp. 1 – 6.

[13] Joel L. Fernandes, Ivo C. Lopes, Joel J. P. C.Rodrigues, Sana Ullah, "Performance Evaluation of RESTful Web Services and AMQP Protocol", Fifth International Conference on Ubiquitous and Future Networks (ICUFN), 2-5 July 2013, pp. 810 – 815. [19] Notable AMQP users, http://www.amqp.org/about/examples, cited 28 Jul 2014. [20] I.Fette, A.Melnikov, "The WebSocket Protocol", RFC 6455, Dec 2011.