International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 05 Issue: 06 | June - 2021 ISSN: 2582-3930

Survey on IOT protocols used for application layer

Siddharth Sharma[#1], Adarsh Agrawal[#2], Prof. Sharadadevi Kaganurmath[*3], Prof .Vanishree K[*4]

[*]Assistant Professor, Department of Information Science and Engineering, RV College of Engineering, Bengaluru, India

[#] Department of Information Science and Engineering, RV College of Engineering,
Bengaluru, India

Abstract **- The standards, technologies, and platforms that make up the IoT ecosystem are rapidly evolving. The Internet of Things (IoT) has emerged as one of the most important computing areas as a result of fast technological developments and the development of internet-connected gadgets. The Internet of Things allows things to communicate and make decisions in a variety of contexts, including healthcare, home automation, disaster recovery, and industry automation. In the future, it is projected to spread to even more uses. While developers are establishing the Internet of Things with current technologies, research organizations are working to adapt protocols to the IoT in order to increase connection.**

The 'Internet of Things' connects commonplace objects or 'things' to the internet via sensors and electronic software, allowing them to interact, gather, and share data. IoT has infiltrated every aspect of our life, including business, healthcare, retail, and security. To maximize communication, a uniform IOT architecture and standardization in IOT protocols are required.

I. INTRODUCTION

During the last decade, the number of devices connected to the Internet has expanded at an unprecedented rate. The Internet of Things (IoT) concept connects not just computers and smartphones to the internet, but also automobiles, dishwashers, televisions, and other common home appliances. In 1999, British entrepreneur Kevin Ashton invented the term “I O T" Since then, the IOT has come a long way. Beyond computing and computer device communication, the internet's breadth is expanding. Its mission is to provide advanced services to society, businesses, and other entities.

The phrases "internet" and "things" make up the IoT. It allows non-computer entities to communicate and coordinate decision-making, allowing them to hear, see, think, calculate, and act. To put it another way, it enables things to act intelligently and reach judgments that benefit a wide range of applications. In computing, communication, and collaboration, they change objects or sensors from passive observers to active collaborators and decision-makers.

Embedded smart sensors, new computing paradigms, data analytics, lightweight communication, and the internet allow the Internet of Things, but it also needs the creation of specific standards and communication protocols to handle the issues that occur. When devices are linked to the internet and require communication channels to communicate with one another, application layer protocols come into play. They're the messaging protocols that IoT devices use to send and receive data; without them, devices wouldn't be able to interact. They provide safe and efficient data transfer, which is essential for IoT devices to function in today's society. Only a handful of the existing application layer protocols mentioned in this book include CoAP, MQTT, XMPP, AMQP, DDS, REST, WebSockets, and JMS.

II. PROTOCOLS

(i) Message Queue Telemetry Transport (MQTT)

MQTT is an Internet of Things communication protocol that has been acknowledged by OASIS (IoT). It's designed to be a super-lightweight publish/subscribe message transport for connecting far- flung devices with minimum coding and network resources. MQTT is presently utilized in a wide range of industries, including transportation, manufacturing, telecommunications, oil and gas, and so on. The MQTT protocol defines two sorts of network entities: a message broker and a group of consumers. An MQTT broker is a server that receives all client messages and routes them to the appropriate destinations. Any device that runs a MQTT library and connects to a MQTT broker via a network is referred to as a MQTT client (anything from a microcontroller to a full-fledged server). The data is organised according to a

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 05 Issue: 06 | June - 2021 ISSN: 2582-3930

subject hierarchy. When a publisher has fresh data to publish, it sends a control message along with the data to the linked broker. The data is subsequently sent to any brokers' clients who have signed up for that topic. The publisher client isn't required to know how many subscribers there are or where they live, and the subscribers don't need to know who the publishers are.

Types of Messages

- Connect - It waits for a connection to be established with the server before establishing a link between the nodes.
- Disconnect -Waits for the MQTT client to complete its task before terminating the TCP/IP connection.
- Publish - Returns to the application thread after broadcasting the request to the MQTT client.

(ii) Extensile Messaging and Presence Protocol (XMPP)

Instant messaging (IM), presence information, and contact list management are all supported by this open communication protocol. It is based on XML and allows for the transfer of structured data in near-real time between any two or more network nodes (Extensible Markup Language). Signaling for VoIP, video, file transfer, gaming, and other applications are among the many uses for the extensible protocol in the wider area of message-oriented middleware.

The application layer of XMPP, unlike most commercial instant messaging systems, is specified by an open standard. In the same way that anybody may operate an email server, anybody may operate an XMPP server, and there is no central master server in the XMPP network. A 'JID' user account, similar to an email address, is used in this federated open system method to allow users to connect with others on any server. Numerous servers, clients, and library implementations are available as free and open-source software, and many XMPP implementations can be produced under any software licence. A wide range of freeware and commercial software implementations are also available. In 2004, the protocols were designed as a recognised instant messaging standard by the eponymous open-source organisation. Other extensions and features have been added since then. XMPP client software is available for both desktop and mobile platforms and devices, according to the XMPP Standards Foundation, and the protocol was used by over ten million people on the network by 2003.

(iii) Representational State Transfer (REST)

"Representational state transfer" (REST) is a software architectural paradigm that uses a subset of HTTP. It's frequently used to build interactive Web services applications. A RESTful Web service abides by a set of guidelines.

For such a Web service to deliver its Web resources in textual format and allow them to be read and changed, it needs a stateless protocol and a set of actions. Computer systems that provide different Internet services can connect with one another using this way. REST is an alternative to SOAP for interacting with Web services.

Web resources were initially characterised as URL-identified documents or files on the World Wide Web. The current definition is much broader and more generic, embracing everything, item, or activity that may be recognised, named, addressed, handled, or performed in any way on the Internet.

Requests to a resource's URI in a RESTful Web service result in a response with a payload organised in HTML, XML, JSON, or another format. For example, the answer might certify that the resource status has changed. In addition, hypertext links to comparable websites might be added in the response. For these requests and answers, HTTP is the most often used protocol. Requests such as GET, POST, PUT, and DELETE are all supported (HTTP methods). RESTful systems employ a stateless protocol and standard procedures to provide high performance, reliability, and scalability by reusing components that can be maintained and updated without impacting the system as a whole, even while it is in use.

Performance, scalability, simplicity, modifiability, visibility, portability, and dependability are all goals of REST. REST concepts including client–server architecture, statelessness, cacheability, layered systems, code-on-demand support, and a unified interface are used to accomplish this. Some conditions must be met in order for a system to be labelled as RESTful.

(iv) Constrained Application Protocol (CoAP)

CoAP is a service layer protocol designed for low-resource internet devices, such as sensor network nodes without cables. CoAP is designed to be readily converted to HTTP for web integration while still addressing certain needs like multicast, low overhead, and simplicity.

Multicast, minimal overhead, and simplicity are required for Internet of Things (IoT) and Machine-to-Machine (M2M) devices, which are frequently embedded and have substantially less memory and power than traditional internet devices. As a result, efficiency is critical. Almost any device that supports UDP or a UDP analogue may utilise CoAP.

The Internet Engineering Task Force's (IETF) Constrained RESTful Environments Working Group (CoRE) has done the majority of the protocol's standardisation work. The protocol has been enhanced with several additional features to make it more appropriate for IoT and M2M applications. The protocol's core is defined by RFC 7252, and substantial additions are in various stages of standardisation.

(v) Advanced Message Queuing Protocol (AMQP)

Anyone can utilise the Advanced Message Queuing Protocol (AMQP), a message-oriented middleware application layer protocol. Message orientation, queuing, routing (including point-to-point and publish-and-subscribe), dependability, and security are all part of the AMQP protocol. AMQP, like SMTP, HTTP, FTP, and other protocols, defines the behaviour of the message provider and client to the point where various manufacturers' implementations are interchangeable.
Previous middleware standardizations (such as JMS) were more concerned with standardising programmer interaction with a variety of middleware implementations than with ensuring interoperability. Unlike JMS, which provides an API and a set of required behaviours for messaging implementations, AMQP is a protocol that runs on the wire.
A wire-level proto- col describes the format of data transmitted across the network as a stream of bytes. As a result, any tool that can send and receive messages that conform to this data format may communicate with any other conforming tool, regardless of implementation language.

III. CONCLUSION

The protocols' characteristics can be summarised as follows:

1. To enable data-centric communication with latency tolerance, MQTT employs the Publish-Subscribe principle.
2. AMQP uses the same Publish-Subscribe architecture as Publish-Subscribe, but it places a greater focus on message-oriented features. Using Exchange-Binding-Queue components, AMQP may queue and route messages in a number of ways.
3. CoAP is a RESTful protocol based on UDP/IP that is designed for low-resource networks. It supports both the Client-Server and Observer Modes, which are analogous to the Publish-Subscribe paradigm.
4. A subset of HTTP is used in representational state transfer, which is a sort of software architecture.
5. XMPP (Extensible Messaging and Presence System) is an open communication protocol that enables instant messaging, presence information, and contact list management.
6.

Because of its minimal overhead and power consumption, MQTT is the most extensively used protocol in IoT. These standards are determined based on the company's and application's requirements. If an application was previously designed using XML and XMPP may be the best option among session layer protocols if it can tolerate a little bit of overhead in its headers.

MQTT, whereas, is the best choice if the application has a large overhead and demands a lot of power.
This, however, is in addition to the deployment of the broker. Because it is HTTP-based protocol.
CoAP would be the greatest, if not the only, choice if the application required REST functionality. The table below summarises the points of comparison between these several session layer protocols.

Protocols	**UDP/TCP**	**Architecture**	**Security and QoS**	**Header Size (bytes)**	**Max Length**
MQTT	TCP	Pub/Sub	Both	2	5
AMQP	TCP	Pub/Sub	Both	8	-
CoAP	UDP	Req/Sub	Both	4	20 (typical)
XMPP	TCP	Both	Security	-	-
DDS	TCP/UDP	Pub/Sub	QoS	-	-

Table 1: Parameters regarding IoT Application Layer Protocols

IV. REFERENCES

[1] Shelby, Z., Hartke, K., & Bormann, C. (2014). The constrained
application protocol ().
[2] Salman, T. Internet of Things Protocols and Standards.
[3] Oh, S., Kim, J. H., & Fox, G. (2010). Real-time performance analysis for publish/subscribe systems. Future Generation Computer Systems, 26 (3), 318-323.
[4] Thangavel, D., Ma, X., Valera, A., Tan, H. X., & Tan, C. K. Y. (2014, April). 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 (pp. 1-6). IEEE.
[5] Talaminos-Barroso, A., Estudillo-Valderrama, M. A., Roa, L. M.rrrr, Reina-Tosina, J., & Ortega-Ruiz, F. (2016). A Machine-to-Machine protocol benchmark for eHealth applications–Use case: Respiratory rehabilitation. Computer Methods and Programs in Biomedicine, 129, 1-11.
[6] Karagiannis, V., Chatzimisios, P., Vazquez-Gallego, F., & AlonsoZarate, J. (2015). A survey on application layer protocols for the internet of things. Transaction on IoT and Cloud Computing, 3(1), 11-17.
[7] Banks, A., & Gupta, R. (2014). MQTT Version 3.1. 1. OASIS Standard.
[8] Lampkin, V. (2012). What is mqtt and how does it work with

websphere mq?
[9] Collina, M., Bartolucci, M., Vanelli-Coralli, A., & Corazza, G. E. (2014, September). Internet of Things application layer protocol analysis over error and delay prone links. In Advanced Satellite Multimedia Systems Conference and the 13th Signal Processing for Space Communications Workshop (ASMS/SPSC), 2014 7th (pp. 398-404). IEEE.
[10] Bendel, S., Springer, T., Schuster, D., Schill, A., Ackermann, R., & Ameling, M. (2013, March). A service infrastructure for the internet of things based on XMPP. In Pervasive Computing and Communications Workshops (PERCOM Workshops), 2013 IEEE International Conference on (pp. 385-388). IEEE.
[11] Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., & Ayyash, M. (2015). Internet of things: A survey on enabling
technologies, protocols, and applications. Communications Surveys & Tutorials, IEEE, 17(4), 2347-2376.
[12] Laine, M. (2012). RESTful Web Services for the Internet of Things. Online] Saatavilla: Http://mediaTkk Fi/webservices/personnel/markku_laine/restful_web_services_f or _the_internet_of_things Pdf.
[13] Rodriguez, A. (2008). Restful web services: The basics. IBM developerWorks.
[14] Bloebaum, T. H., & Johnsen, F. T. (2015, October). Evaluating publish/subscribe approaches for use in tactical broadband networks. In Military Communications Conference, MILCOM 2015-2015 IEEE (pp. 605-610). IEEE.
[15] Fernandes, J. L., Lopes, I. C., Rodrigues, J. J., & Ullah, S. (2013, July). Performance evaluation of RESTful web services and AMQP protocol. In Ubiquitous and Future Networks (ICUFN), 2013 Fifth International Conference on (pp. 810-815). IEEE.
[16] Standard, O. A. S. I. S. Oasis advanced message queuing protocol (amqp) version 1.0., 2012. URL http://docs. Oasis-open. org/amqp/core/v1. 0/os/amqp-core-complete-v1. 0-os. Pdf.
[17] C. Bormann, A. P. Castellani, Z. Shelby, "CoAP: An application protocol for billions of tiny Internet nodes", IEEE Internet Comput., vol. 16, no. 2, pp. 62-67, Mar./Apr. 2012.
[18] C. Lerche, K. Hartke, M. Kovatsch, "Industry adoption of the Internet of Things: A constrained application protocol survey", Proc. IEEE 17th Conf. ETFA, pp. 1-6, 2012.
[19] Davis, Ernesto, Calveras, Anna & Demirkol, Ilker 2013, 'Improving packet delivery performance of publish/subscribe protocols in wireless sensor networks', Vol. 13, No. 1. Multidisciplinary Digital Publishing Institute, 2013, pp. 648–680.
[20] De Caro, N, Colitti, W, Steenhaut, K, Mangino, G & Reali, G 2013, 'Comparison of two lightweight protocols for smartphone-based sensing', Communications and Vehicular Technology in the Benelux (SCVT), 2013 IEEE 20th Symposium On, pp. 1-6.
[21] Colitti, W, Steenhaut, K, De Caro, N, Buta, B and Dobrota, V 2011,'Evaluation of constrained application protocol for wireless sensor networks,' Local & Metropolitan Area Networks (LANMAN), 2011 18th IEEE Workshop On, 2011, pp. 1-6.
[22] Thamer, Alghamdi, Aboubaker, Lasebae & Mahdi, Aiash 2013,' Security Analysis of the Constrained Application Protocol in the Internet of Things', Second International Conference on Future Generation Communication
Technologies (FGCT), London, UK.
[23] Talaminos-Barroso, A, Estudillo-Valderrama, M. A., Roa, L. M.rrrr, ReinaTosina, J., & Ortega-Ruiz, F 2016, 'A Machine-to-Machine protocol benchmark for eHealth applications–Use case: Respiratory rehabilitation', Computer Methods and Programs in Biomedicine, 129, 1-11, 2016.
[24] Ananya, Pramanik, Ashish, Luhach, Isha, Batra & Upasana, Singh 2017,' A Systematic Survey on Congestion Mechanisms of CoAP Based Internet of Things', Advanced Informatics for Computing Research, July 2017, pp 306- 317.