American Journal of
Electronics & Communication

Rohit Kumar Thakur et. al., American Journal of Electronics & Communication, Vol. III (1), 28-34

A Comparison of Various IoT Application Layer Protocol

Rohit Kumar Thakur
Information Technology University Institute of
Engineering and technology
Panjab university
Chandigarh, India
thakur.rohit1997@gmail.com

Raj Kumari
Information Technology University Institute of
Engineering and technology
Panjab university
Chandigarh, India
rajkumari@puchd.ac.in

Abstract

Data communication over the internet is highly swayed by protocols. So, the protocols have a very crucial role in data communication. Similarly, for the Internet of things (IoT) enabled devices the protocols have also an important role. In the Application-layer of IoT-enabled devices, the protocols have a significant role in the transfer of data from server to IoT devices and vice versa. There are different protocols are there, some are taken from web based internet protocols like Hyper Text Transfer Protocol (HTTP), Transmission Control Protocol (TCP), User Datagram Protocol (UDP), etc. And some are exclusively developed for IoT devices like Message Queue Telemetry Transport (MQTT), Constraint Application Protocol (CoAP), etc. IoT-enabled devices need a lightweight, secure and fast protocols for the transfer of data. When IoT enabled devices are developed there are limited number of protocols developed for communication of IoT devices. Various protocols are developed to fulfil the need of users but they have some advantages as well as disadvantages. In this paper, analysis of the performance and comparision of these protocols through different scenarios are given.

Keywords: Internet of Things, HTML, CoAP, MQTT, XMPP, Web Socket;

I. Introduction

The IoT encompasses everything that connects to the internet, including smartphones, tablets, desktops, and laptops. However, the term is often used in a slightly narrower sense. The "things" being referred to objects that can talk to each other i.e smart speakers, plugs, lights, heating systems, fridges, cars, etc. Devices like smartphones and computers are already internet-connected [1]. IoT technology consists of devices that can be operated over the internet. Nowadays, IoT technology covers mostly all the fields like smart homes, agriculture fields, and medical science. IoT is essential for every person in society. The importance of IoT can be understood by a simple example of a smart home where electric equipment is controlled by IoT sensors that switch on/off according to need. This smart home system saves electricity. And this electricity can be used somewhere in the rural area for the irrigation of fields. Here the importance of IoT devices in the common man's life can be understood. Some of IoT enabled devices use Real Time Application (RTA) to access data and also for monitoring purposes. RTA is called real-time computing and this is done with the help of IoT technology. A recent study shows that over 20 million IoT devices are connected over the internet to perform various tasks in different fields with real-time computing [2].

The IoT enabled devices have five layers i.e., perception layer, network layer, middleware layer, application layer, business layer. These layers perform a crucial role during the connectivity of IoT devices. The perception layer is used for physical objects like sensors. The network layer is used for transmission. The middleware layer is used for storage, information processing, and actions. The application layer is used for smart application and management. The Business Layer is used for analytics purposes.

Every technology has some issues and limitations, similarly, IoT technology has. As the storage capacity is limited in IoT enabled devices so security mechanisms can not be implement efficiently. Data is valuable for each user and hence for its security, a strong protocol is required. One more is Quality of service (QoS) which means fast and error-free transmission of data. Interoperability is also there as IoT enabled devices uses different configurations in different devices according to the company which makes a hurdle in communication. The communication between different IoT-enabled devices is poor due to the issue of interoperability [3].

This paper discusses all the challenges at the application layer and above mentioned issues are solved by using different protocols. The study of application layer protocols how they solve challenges which are discussed above like security, QoS and Interoperability. The comparison between different application layer protocols is discussed here and how one protocol is better than the others. This comparison is done on various parameters like QoS, light weight, speed of communication and error handling etc.

.

II. Related Work

IoT protocols has been developed very fast and efficiently since the last decade. Some important research in

IoT protocols is in the speed and more lightweight nature of the protocols. In [4], authors have elaborated various aspects of lightweight cryptography. Authors have proposed a lightweight protocols for IoT devices. For security refer to [5] where authors elaborated various security aspects while using IoT devices over the internet. Visions and challenges are discussed by research scholars in [6, 7, 8].

Handling of various data protocols like XMPP, CoAP, AMQP, MQTT, DDS and MQTT-SN of IoT is discussed in [9]. The working of these protocols is explained thoroughly in [10]. For performance refer to [11] which explain performance of IoT enabled devices.

From the last three decades, HTTP has been used as a mainstream protocol for data transfer [12]. It has various advantages and disadvantages explain in this paper. Various application layer protocols like CoAP, MQTT, XMPP and WebTransport protocol are also explained. Comparison of these different application layer protocols on different standards and how they overcome these problems are discussed. Analysis of various situation and on these situation which protocol is suitable are also explained.

Protocols which are new in the technology like MQTT and XMPP protocol are advanced as compared to an older one. The researchers are still working on these protocols to make them more efficient and lightweight.Various measures and standards such as lightweight, security, interoperability, speed etc. are taken in this paper to compare all these protocols. Comparison between these protocols is done very efficiently based on previous researches done by different research scholars.

III. Components of an IoT system

Various components of IoT [2] are shown in Fig. 1. The details of each component is discussed below.

D. *IoT Edge Devices*

IoT edge devices form the smart IoT actuator since they are able to conduct some processing themselves. Cloud computing and IoT have elevated the role of edge devices , ushering in the need for more intelligent, computing power and advanced service at the network edge.

E. *IoT Sensors*

All IoT-enabled devices need to have one or more sensors to collect data from the environment. These sensors are connected to the cloud, where they can transmit and receive data. IoT sensors are mostly small in size, have low cost and consume less power.

F. *Device Provision*

Every data provision service instance is assigned a so-called ID scope on creation which is unique and never change during the whole lifetime of the instance. It helps a large number of devices to be connected and registered.

G. *IoT gateway/framework*

A IoT gateway framework is a software solution that bridges the semantic gap between the raw sensor data and the information context that is received by a high level application. It collects data from IoT sensors and edge devices and transfers that data over the cloud server.

H. *Stream Processing*

IoT enabled devices have ability to collect , integrate , analyze and visualize continuous data streams in real time called stream processing. So data from the cloud server is collected by stream processing and various operations are performed on data as discussed above and then transferred to the user interface.Stream processing in IoT is for processing data to generate information according to user need.

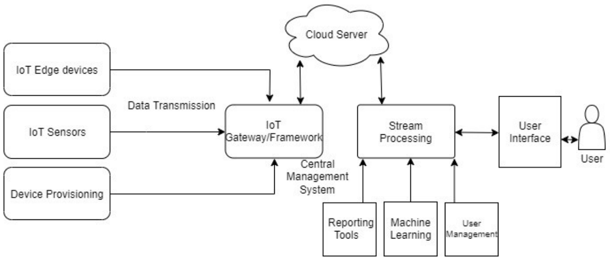

Fig. 1. IoT Architecture [2]

I. *Reporting Tools*

These tools help to hold and store data, while provide necessary tools for batch processing. Some examples of tools are Open-source Hadoop frameworks such as Spark and MapReduce are a popular choice for big data and for smaller data sets and application data, you might use batch ETL tools such as Informatica and Alteryx.

J. *Machine Learning and User Management*

IoT sensors generate massive volume of data. Machine Learning is actuated by data and generates perception for it. Machine Learning uses previous information to identify patterns and builds models that help predict future behaviour and events. So in IoT machine learning is used to handle with real time reaction of sensors toward the situation. For the management of various resources available in the system user management is used.

K. *User Interface*

The characteristics via which a user interacts with an IoT system are referred to as the user interface. This covers things like screens, pages, buttons, icons, and forms. Software and programmes on computers and smartphones are the most visible examples of user interfaces.

Ⅳ. Protocols in Application Layer

In IoT architecture, various protocols are used at the application layer named as Message Queue Telemetry Transport (MQTT), Advanced Message Queuing Protocol (AMQP), Extensible Messaging and Presence Protocol (XMPP), Hyper Text Transfer Protocol (HTTP), Constraint Application Protocol (CoAP) and Websocket. The detail discussion on these protocols which are used for communication at the application level is as follows:

A. Hyper Text Transfer Protocol(HTTP)

This paper discussed HTTP and its best alternative according to various aspects like security, communication speed, and resource management. In HTTP protocol [13], it uses Transmission Control Protocol (TCP) for the transfer of data from one system to another system. It is early developed for the text-based transfer of data from server to client and client to server. TCP which is used for sending data is a “connection” oriented protocol that works on the client-server model as shown in Fig 2, the client generates a request, and the server responds to the request. HTTP is big size protocol and it has long header size which is developed for web-based communication. Problems arise in IoT technology while using it on IoT platforms.These problems and some advantages are explained below in subsequent section.

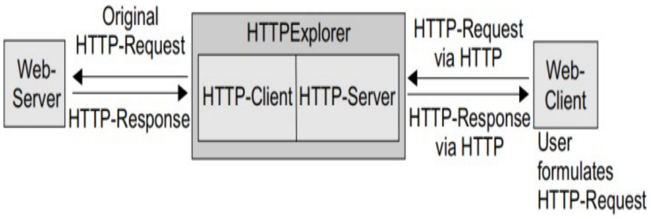

Fig. 2. HTTP protocol Architecture [14]

Advantages

- HTTP has the advantage of staying connected for a short time while the device is sending and receiving data, but MQTT which is discussed in next section needs to stay connected. HTTP (if used correctly) may be able to handle more traffic, as some systems can only handle some specific number of connections at one time initially handle more traffic [8].
- HTTP reports errors without closing the TCP connection.
- HTTP uses TCP which is more reliable than CoAP which is discussed next section.[13]
- Handshaking is done at the initial connection establishment stage. Hence it offers reduced latency in subsequent requests as there is no handshaking.

Problems

- HTTP connects only two systems at a time while in IoT technology we need to connect and control devices whose counting may be more than two to thousands. Like in heavy industries there are hundreds of sensors that are connected so HTTP is not suitable to communicate with multiple devices at the same time.[13]
- HTTP used in IoT technology they request resources from the cloud or server and it takes time to respond to the request as a maximum of IoT are small in size and very limited computing resources most of them are based, asynchronous models.
- In IoT Telemetry and Telecommand are to be executed at the same time. But HTTP is based on the Request-Response model so it is very tough to implement this on IoT devices as we need to send data on both sides.[14]
- HTTP consumes and uses very high power as now today's advanced wireless sensor network which has small batteries that are not fit for HTTP so we need a new alternative for this to increase the performance of IoT devices that use sensors.
- HTTP only sends data when requested by the client. When the data is requested it will be sent to the user otherwise there will be no transfer of data.
- HTTP is not suitable in resource constrained environments because it is fairly verbose in nature and thus, incurs a large parsing overhead.

B. Constraint Application Protocol(CoAP)

Among the others, such as Message Queue Telemetry Transport (MQTT) or Advanced Message Queuing Protocol, the Constrained Application Protocol (CoAP) [15] takes the lead (AMQP). This protocol is lightweight and can run on devices and networks with little resources, and it can be secured using Datagram Transport Layer Security (DTLS). Because IoT devices interact with the physical world and transmit personal data, having a secure communication channel is critical in IoT contexts.The architecture of CoAP is shown in Fig. 3 where various client and server are connected to each other. Clients are connected through CoAP and Server are connected through HTTP protocol. CoAP is lightweight but not so much secure than HTTP. As it is similar to HTTP, only compromise in some parameters like security. Some advantages and disadvantages of CoAP are as discussed below.

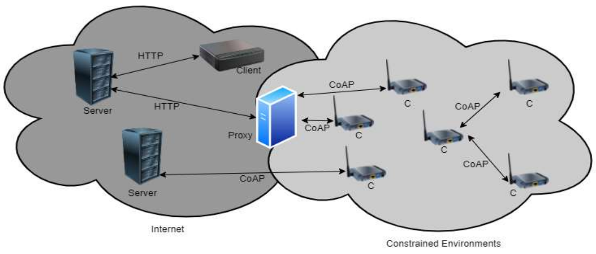

Fig. 3. CoAP Protocol Architecture [16]

Advantages

- It operates over User Datagram Protocol (UDP), which requires minimal overhead for communications. It also allows faster wake up times and extended sleepy states. Taken together, this means batteries last longer for IoT devices [16].
- Another advantage of UDP is small packet sizes. This leads to faster communication cycles. Therefore, CoAP allows batteries to last longer.
- In CoAP, Datagram Transport Layer Security (DTLS) is employed over UDP, communication is encrypted and secure.[17]
- CoAP uses Efficient XML interchanges(EXI) date format and is far more efficient in term of space as compare to plain text HTML/XML
- CoAP also support feature of header compression resource discovery, auto configuration, asynchronous message exchange, congestion control and support multicast in CoAP.
- In CoAP, the reliable transmission of data over UDP is done using confirmation message, which increases the accuracy [16].

Problems

- It is a subset extension of HTTP, therefore, it is not compatible while making multiple connections. It is also unidirectional in nature and based on the client-server model.[16]
- Security is also a major concern in CoAP.It doesn't have a reliable standard for security architecture.[17]

C. *Message Queue Telemetry Transport (MQTT)*

The MQTT protocol [18] has been proclaimed as "the protocol" for the IoT by the open standards body, OASIS, and a major technology company, IBM. It's been marketed as a low-power alternative to HTTP and other Internet-of-Things protocols (Constrained Application Protocol - CoAP, Advanced Messaging Queuing Protocol - AMQP, etc.). It was created to design a protocol for linking oil pipelines through a satellite connection with the least amount of battery loss and the least amount of bandwidth. Its goals were to be easy to implement a protocol that provided Quality of Service, Data Delivery, bandwidth-efficient and data-agnostic while maintaining continuous "session awareness" for collecting information of ongoing session. It is lightweight and can be easily implemented in resource-constrained devices like temperature or pressure sensors, light bulbs etc. It solves the one-to-many problem that many other technologies struggle to implement. Architecture of MQTT is shown in the Fig 4. where various client are connected to MQTT broker client may be a sensor or any user which is using the data of sensors using broker. Sensor can only publish while for any system who is getting the data first subscribe the MQTT broker then it publish the information [19]. Advantages and disadvantages are explained in subsequent section.

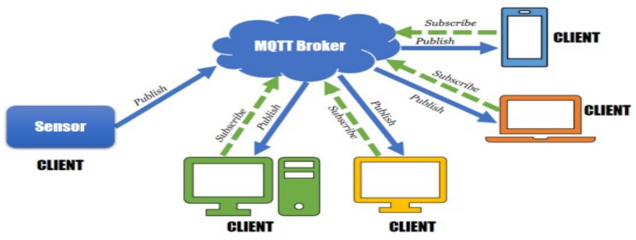

Fig. 4. MQTT Protocol Architecture [19]

Advantages

- It provides data security with SSL/TLS which is not provided by HTTP.
- MQTT uses TCP and hence perfect for scenarios where connectivity is required all through, but CoAP uses UDP which ensures effective battery consumption through the connectionless model.
- MQTT allows to keep a message on the broker for an indefinite amount of time. Fresh clients that subscribe to that specific topic will receive an immediate message with the most recent good data released, rather than having to wait for new massages from the publisher [20].
- MQTT has been specially developed for IoT devices. Its design principles are to use limited network bandwidth and provide assurance of delivery which are provided by websocket and other protocol [20].

Problems

- MQTT doesn't support interoperability means MQTT transfers data as a byte array. Data is stored in the packet. There is no universally approved encoding for data formatting, and no universally accepted technique for communicating that encoding to the subscriber. There is no interoperability if the sensing device sends data to the display device in a format that the display device does not support.MQTT subscribers are unaware of status changes in the producer. If any producer changes the data or repeats a message MQTT broker will send it to all the subscribers even if it is a duplicate of an earlier message. This is just the wastage of bandwidth used for sending duplicate data.[21]
- MQTT supports asynchronous messaging; whereas, CoAP supports both synchronous and asynchronous messaging [20].
- MQTT requires TLS for security MQTT. It is a TCP Application layer protocol and relies on TLS to provide encryption and security. Unfortunately, by adding TLS, MQTT is no longer a lightweight or low resource application.

D. *Websocket*

A WebSocket is a full-duplex (bidirectional), low-latency (real-time), long-running (persistent), single connection protocol [22] between a client and server. The architecture of Websocket is shown in Fig 5 which shows various components involved in Websocket protocol such as browser which request data and web socket server which update information to the gateway, data storage etc. WebSockets are extremely useful for event-driven, real-time web applications. It's utilized for real-time data synchronization and updates, live text chat, video conferencing, Voice over Internet Protocol (VOIP), Internet of Things (IoT) control, and monitoring. These capabilities enable applications in gaming, social networks, logistics, finance, home, vehicle and industrial automation. Most major web browsers currently support their use. As CoAP and HTTP both are not bidirectional, to overcome this problem Websocket is used. This is compatible with constant data exchange between two IoT devices which was not supported by CoAP and HTTP and it is also low power consumption and lightweight protocol than HTTP [23]. Advantages and disadvantages are discussed below in subsequent section.

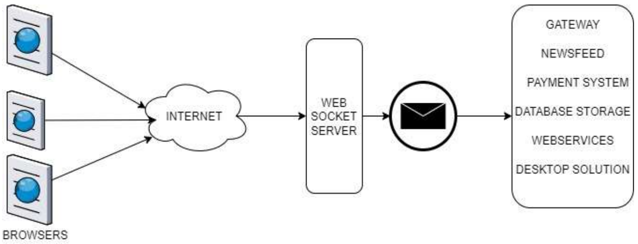

Fig. 5. Websocket Protocol Architecture[25]

Advantages

- It removes the problem of time delay as it is having asynchronous communication. This protocol overcomes this limitation of HTTP.[22]
- Websocket uses a custom binary farming format that divides each message into one or more frames. These frames get combined when they reach the destination. While in HTTP additional 500-800 bytes of metadata plus cookies are used which makes it heavy to transfer.
- Data can be sent to any end without the request while in HTTP data can only be sent only when it is requested by that end.
- WebSocket is an event-driven protocol which can be used for real time communication. In Websockets, updates are sent immediately when they are available.[23]
- It keeps a single, persistent connection open while eliminating latency problems that arise with HTTP request/response-based methods.

L. Problems

- Data security is the main issue while using Websocket as it allows the establishment of number of connections.
- It can reduce the metadata (HTTP headers) that are sent in every request and it also provide full–duplex communication through a single connection [24].
- It does not fulfill all the security standards. The two available options with WebSockets are either WS(WebSocket) or WSS(WebSocket Secure).

E. Extensible Messaging and Presence Protocol(XMPP)

XMPP [26] is an open set of rules for exchanging messages and presence information in near-real-time using streaming XML components. As seen in Fig. 6, the XMPP protocol is based on a standard client-server architecture. Its client connects to an XMPP server using a TCP socket. Beyond standard instant messaging (IM) and the delivery of presence data, XMPP provides a broad foundation for communications across a network, with a variety of uses [19]. It enables the discovery of services located locally or across a network, as well as the determination of their availability.. Various advantages and disadvantages are discussed below.

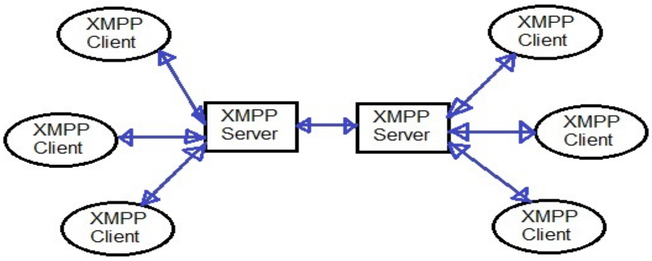

Fig. 6. Architecture of XMPP[19]

Advantages

- It provides both identification of devices and encryption of data. XMPP makes use of two types of encryption methods i.e., SASL (Simple Authentication Security Layer) and TLS (Transport Layer Security) which makes it more secure than other protocol. Here TLS protocol specifies the type of certificate that must be exchanged between nodes while SSL provides keyed message authentication [27].
- It uses short messages for fast communication between user and server.
- It is asynchronous, which means it provide faster update than HTTP.
- It allows servers with different architectures to communicate.
- It inherently supports the publish/subscribe architecture that is more suitable for the IoT[28].
- XMPP has good openness and scalability. It can be used to implement the interoperability between wide varieties of instant messaging systems[29].
- XMPP publish/subscribe scheme has two benefits for energy saving. Firstly, the server will maintain and manage the publish/subscribe relationships between publishers and subscribers, so that the publishers only need to care about whether their data is subscribed and publish each data to the server, the server will broadcast the data to all the authorized subscribers [28].

Problems

- Text-based messaging and no provision for end-to-end encryption in XMPP. Due to this, security issue hinders it while using it in IoT environment.[19]
- It lacks the QoS mechanism used in the MQTT protocol.
- The transfer of XML content is asynchronous.
- Instant messaging is there so server may overload with it.

V. Chart for comparison of different protocol HTTP , CoAP , MQTT , XMPP , WEBSOCKET:

Protocols	Mode of Communication	Size	UDP/TCP	Security and QOS	Header Size	Security Mechanism	REST ful
HTTP	Half Duplex	Heavy Weight	TCP	BOTH	SERVER DEPENDENT	SSL/TLS	Yes
CoAP	Half Duplex	Light Weight	UDP	QOS	4	DTLS	Yes
MQTT	Full Duplex	Light Weight	TCP	BOTH	2	SASL/TLS	No
WEBSOCKET	Full Duplex	Light Weight	BOTH	QOS	-	SSL/TLS	No
XMPP	Full Duplex	-	TCP	SECURITY	-	SASL/TLS	Yes

VII. CONCLUSION:

The present study compared all updates and currently used in IoT application layer protocol HTTP, CoAP, MQTT, Websocket and XMPP. Their implementation and how they are better from each other are compared furthermore Research must be required on these topics and they have various limitations. No protocol is good enough on all standard, some give better QoS and some give better security. We have to compromise with security and where the QoS is high and vice versa. XMPP is better for security as it provide two level of security still a lot of work is required in it. While if the requirement is for lightweight protocol then CoAP and MQTT is best choice from our comparison. For real time update we can use websocket protocol. There is lots of scope possible for research in this field to develop an IoT-specific protocol that provides all the services with no compromise on security and QoS.

References

[1] Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT): A vision,architectural elements, and future directions. Future generation computer systems, 29(7), 1645-1660.

[2] Li, S., Da Xu, L., & Zhao, S. (2015). The internet of things: a survey. Information Systems Frontiers,17(2), 243-259.

[3] Guo, B., Zhang, D., Wang, Z., Yu, Z., & Zhou, X. (2013). Opportunistic IoT: Exploring the harmonious interaction between human and the internet of things. Journal of Network and Computer Applications, 36(6), 1531-1539.

[4] Singh, S., Sharma, P. K., Moon, S. Y., & Park, J. H. (2017). Advanced lightweight encryption algorithms for IoT devices: Survey, challenges and solutions. Journal of Ambient Intelligece & Human Computing.

[5] Koivu, A., Koivunen, L., Hosseinzadeh, S., Laurén, S., Hyrynsalmi, S., Rauti, S., & Leppänen, V.(2016, December). Software Security Considerations for IoT. In Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), 2016 IEEE International Conference on(pp. 392-397). IEEE.

[6] Sundmaeker, H., Guillemin, P., Friess, P., & Woelfflé, S. (2010). Vision and challenges for realising the Internet of Things. Cluster of European Research Projects on the Internet of Things, European Commision, 3(3), 34-36.

[7] Vermesan, O., Friess, P., Guillemin, P., Gusmeroli, S., Sundmaeker, H., Bassi, A., ... & Doody, P.(2011). Internet of things strategic research roadmap. Internet of Things-Global Technological and Societal Trends, 1(2011), 9-52.

[8] Sheng, Z., Yang, S., Yu, Y., Vasilakos, A., Mccann, J., & Leung, K. (2013). A survey on the ietf protocol suite for the internet of things: Standards, challenges, and opportunities. IEEE Wireless Communications, 20(6), 91-98.

[9] M. Anusha, E. S. Babu, L. S. M. Reddy, A. V. Krishna and B. Bhagyasree,"Performance Analysis of Data Protocols of Internet of

Things:Qualitative Review," International Journal of Pure and Applied Mathematics, vol. 115, no. 6, pp. 37-47, 2017.

[10] "SPDY: An experimental protocol for a faster web," last visited 24 February 2021. [Online]. Available: https://dev.chromium.org/spdy/spdy-whitepaper.

[11] "Performance Analysis of Data Protocols of Internet of Things: Qualitative Review," International Journal of Pure and Applied Mathematics, vol. 115, no. 6, pp. 37-47, 2017

[12] I. Paterson and D. Smith D. Saint-Andre and J. Moffitt. XEP0124: Bidirectional-streams Over Synchronous HTTP (BOSH). http://xmpp.org/extensions/xep-0124.html.

[13] Perry Lea, "Internet of Things for Architects", Packt Publishing, 2018.

[14] Volker Turau,"HTTPExplorer: Exploring The Hypertext Transfer Protocol",Technische Universität Hamburg,Conference Paper · January 2003.

[15] A. Kerenen M. Koster and J. Jimenez. Publish-Subscribe Broker for the Constrained Application Protocol. Rfc, RFC Editor, July 2017.

[16] Fathia Ouakasse,Said Rakrak,"An Improved Adaptive CoAP Congestion Control Algorithm",International Journal of Online and Biomedical Engineering (iJOE) · February 2019.

[17] Chang-Seop Park ,"Security Architecture for Secure Multicast CoAP Applications",IEEE INTERNET OF THINGS JOURNAL, VOL. 7, NO. 4, APRIL 2020.

[18] A. Stanford-Clark and H. Linh Troung. MQTT For Sensor Networks (MQTT-SN) Protocol Specification Version 1.2. Mqtt.Org, 2013.

[19] Neven Nikolov,"Research of MQTT, CoAP, HTTP and XMPP IoT Communication protocols for Embedded Systems"XXIX International Scientific Conference Electronics - ET2020, September 16 - 18, 2020

[20] Biswajeeban Mishra,Attila Kertesz,"The Use of MQTT in M2M and IoT Systems: A Survey",IEEE Access November 4, 2020.

[21] Dipa Soni,Ashwin Makwana,"A SURVEY ON MQTT: A Protocol of Interner of Things(IOT)",Researchgate Conference Paper, April 2017.

[22] D. Sheiko, "Persistent Full Duplex Client-Server Connec-tion via Web Socket," 2010.http://dsheiko.com/weblog/persistent-full-duplex-client-ser-ver-connection-via-web-socket.

[23] Makoto, "Living on the Edge of the WebSocket Proto-col," 2010.http://blog.new-bamboo.co.uk/2010/6/7/living-on-the-edgeof-the-websocket-protocol.

[24] Bhumij Gupta1, Dr. M.P. Vani2,"An Overview of Web Sockets: The future of Real-Time Communication",International Research Journal of Engineering and Technology (IRJET),Volume: 05 Issue: 12 | Dec 2018.

[25] Chandadeo Kumar, Rajak Umang Soni, Biswajit Biswas, A.K Shrivastava,"Real-time web based Timing display Application for Test Range Applications",2nd International Conference on Range Technology (ICORT)2021.

[26] A. Hornsby and R. Walsh. From instant messaging to cloud computing, an xmpp review. In IEEE International Symposium on Consumer Electronics (ISCE 2010), pages 1–6, June 2010.

[27] Halil Arslan,"Extensible messaging and presence protocol's adaptation to business applications ",Global Journal of Computer Sciences: Theory and Research Volume 06, Issue 1, (2016) 10-17.

[28] Heng Wang; Daijin Xiong; Ping Wang; Yuqiang LiuA(2017).Lightweight XMPP Publish/Subscribe Scheme for Resource-Constrained IoT Devices

[29] B. Xuefu and Y. Ming, "Design and implementation of Web instant message system based on XMPP", Proc. IEEE 3rd Int. Conf. Softw. Eng. Service Sci. (ICSESS), pp. 83-88, Jun. 2012.