The ISC Int'l Journal of Information Security
ISeCure

August 2019, Volume 11, Number 3 (pp. 173–180)
http://www.isecure-journal.org

Selected Paper at the ICCMIT'19 in Vienna, Austria

IoT Protocols Based Fog/Cloud over High Traffic

Istabraq M. Al-Joboury [1,*], and Emad H. Al-Hemiary [1]
[1] *Networks Engineering Department, College of Information Engineering, Al-Nahrain University, Iraq*

ARTICLE INFO.

Keywords:
Internet of Things, Web of Things, Fog Computing, Cloud Computing, Edge Computing, MQTT, HTTP, Tsung.

Abstract

The Internet of Things (IoT) becomes the future of a global data field in which the embedded devices communicate with each other, exchange data and making decisions through the Internet. IoT could improve the quality of life in smart cities, but a massive amount of data from different smart devices could slow down or crash database systems. In addition, IoT data transfer to Cloud for monitoring information and generating feedback that will lead to high delay in infrastructure level. Fog Computing can help by offering services closer to edge devices. In this paper, we propose an efficient system architecture to mitigate the problem of delay. We provide performance analysis like response time, throughput and packet loss for MQTT (Message Queue Telemetry Transport) and HTTP (Hyper Text Transfer Protocol) protocols based on Cloud or Fog servers with large volume of data from emulated traffic generator working alongside one real sensor . We implement both protocols in the same architecture, with low cost embedded devices to local and Cloud servers with different platforms. The results show that HTTP response time is 12.1 and 4.76 times higher than MQTT Fog and Cloud based located in the same geographical area of the sensors respectively. The worst case in performance is observed when the Cloud is public and outside the country region. The results obtained for throughput shows that MQTT has the capability to carry the data with available bandwidth and lowest percentage of packet loss. We also prove that the proposed Fog architecture is an efficient way to reduce latency and enhance performance in Cloud based IoT.

1 Introduction

IoT is a new concept and paradigm; in which the real worlds of things linked to the virtual world, subsequently enabling anything to anyone [1]. It is becoming a revolution of devices as well as representing the future of the Internet. Therefore, it has attracted wide attention from researchers. The IoT technology consists of two terms "Internet" and "Things". The first term gives the meaning of protocols, services and networks, whereas the second term refers to sensors, smart devices [2]. The basic idea of IoT is that objects (such as electronics) connected together to

* Corresponding author.
Email addresses: estabriq_94@coie-nahrain.edu.iq, emad@coie-nahrain.edu.iq
ISSN: 2008-2045

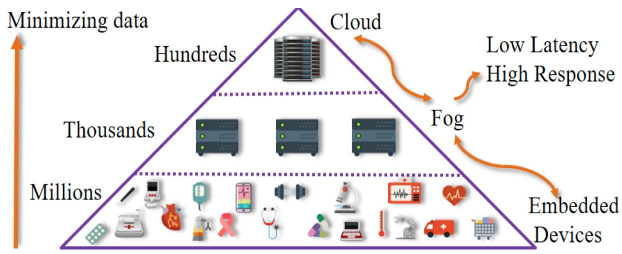

Figure 1. Cloud, Fog and embedded devices layers

provide an efficient, low power, and seamless connectivity to humans [3]. Then, new technologies allow objects to be more intelligent, which can transfer data generated from different things, as well as, make them recognizable by using IP and RFID. This leap leads to the integration of IoT and Cloud computing. Furthermore, IoT moves into the unlimited capabilities of IP6 addresses [4]. The elements of IoT can be a physical/digital entity, which perform various daily tasks for individual users and then IoT application protocols and technologies used to achieve IoT vision; for instance, wireless sensor networks allow objects to measure in real time and data is collected by using IoT protocols. Smart city provides services to governments (e.g smart transportation and mobility, smart building and infrastructure, organizations (e.g e-learning, manufacturing, smart factories) and humans (e.g smart home, smart hospital). The common IoT layers are three model categorizing Application, Network, and Perception Layers. new layers: Business and Middleware Layers are recently proposed [5].

Fog Computing is a concept made by Cisco in 2012, that aiming the real time applications to handle billions of IoT devices [6]. It refers as an intermediate layer between Cloud and embedded devices enabling storage, computing and networking services, the same as Cloud Computing. It consists of servers, routers, switches and access points [7]. Fog Computing brings all based Cloud features and services near to edge devices "ground" like sensors, smartphones, wearables and embedded devices [8] as shown in Figure 1. Smart cities including smart hospitals and infrastructures for IoT environments that they handle big data and stream for real time application. Thereby, a real time city is enabled due to offer new services for governments and societies, as well as big data analysis in real time of infrastructure level and the person's lifestyle. Here, data generated from IoT devices sent to the Cloud in order to be stored and processed. Cloud computing enables services (Software as a Service - SaaS) and (Infrastructure as a Service – IaaS) and Platform as a Service -PaaS) and provides data processing. It is suitable for applications that their data is stored and processed in centralized. Some application such as health care systems depends on distributed storage

Table 1. Comparison between Fog and Cloud [10]

	Fog	Cloud
Location	Local	Internet
Data	Thousands	Hundreds
Latency and Delay	Low	High
Storage	Distributed	Centralized

and low latency, at this point Cloud fails to handle these conditions [9]. However, there are some differences between the two concepts as it discusses briefly below in Table 1.

The size of packet contents of HTTP and MQTT protocols from sensors using open source network analyzer Wireshark as is shown in Table 4. IPerf is used as a tool to measure network bandwidth between sensors and Fog, and between sensor and Cloud (located in Ministry of Higher Education and Scientific Research). IPerf is a powerful and simple testing tool, client/server model written in C++, it used to analyze performance network quality, loss and bandwidth based on TCP or User Datagram Protocol (UDP). Table IV summarizes the correlation between location of servers and ISP (Internet Service Provider) based on available bandwidth. The proposed IoT architecture consists of integrated the simulation and practical work. Each of (mosquitto, MongoDB and LAMP) of Fog/Cloud based are installed and configured on HP ProLiant 380 G7 for Fog server and on HP ProLiant 380 G8 for Cloud server, OS: Ubuntu server 14.04 LTS, RAM: 32 GB, processor: 32 and 500 GB. Tsung was installed on different machine with characteristics: OS: Ubuntu 14.04.5 LTS, Memory: 3.7 GB, processor: Intel(R) Core(TM) i3-380 CPU @ 2.53GHz *4, disk: 488.1 GB.

2 Overview of WEB Protocols

The MQTT is an application layer protocol designed for lightweight M2M (machine to machine) communications, simple, easy to implement and fast transportation protocol. MQTT is suitable for resource constrained devices, low bandwidth, low latency and reliable networks. Stanford-Clark and Nipper [11] release the first version of MQTT protocol in 1999; IBM originally created it. The latest version of MQTT is 3.1.1 [Nov, 2014] and it is becoming an open standard protocol. MQTT is an OASIS (Advancing Open Standards for the Information Society) runs over TCP/IP protocol. It is publish/subscribe model based on topics and consists of three elements: two types of clients (publisher or subscriber) and one server (called broker). Publishers send messages within a specific topic, then subscriber clients receive these messages that refers to the same topic that they subscribed via broker as shown in Figure 2. Also, publisher do not require the address of the subscribers [12, 13].

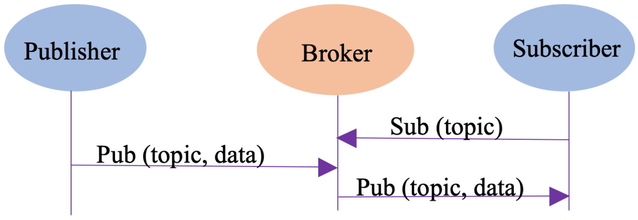

Figure 2. The operation of MQTT Based on Publish/Subscribe Model [14]

MQTT has a lower overhead, a synchronous and reliable with some multiple different levels of quality of services. There are three types of QoS for a delivery assurance that are used between client and server [11, 15]. There are:

- QoS level 0: the publisher sends the message to the subscriber through the broker and the subscriber receives the message at most once. In addition, the broker never sends an acknowledgement to the publisher.
- QoS level 1: the publisher delivers the message to the clients at least once, and the broker send back an acknowledgement if the message is lost.
- QoS level 2: publisher uses level 2 when message lost or duplicate and this requires four-way handshake to deliver the message at exactly once, hence this cause increase in the overhead for this reason level 2 is not included in this paper.

The broker may require authentication (username and password) from subscribers to allow them to connect, so that the broker will confirm the privacy by using (Secure Sockets Layer -SSL)/ (Transport Layer Security-TLS). HTTP is an application layer protocol based on TCP/IP suite of protocols. It used to transfer data from client side like smart phone, personal computer to server side such a Web server over the World Wide Web. HTTP v2. is last version [May 2015] [16]. The most commands are GET and POST for processing data on web. It is request/response model based on Uniform Resource Locators (URLs) the user request data on web server, then server not only response to data but all relevant data to that request. There are some differences between the two protocols as it summarizes below in Table 2.

3 Problem Description

Thousands of sensors that it can be useful for monitoring and analyzing. However, an unprecedented volume of data can crash storage systems and real time applications. Cloud Computing could provide storage "on demand" and processing of systems, but Cloud could be anywhere and away far from systems, as well as transferring data from sensors to Cloud and then giving a feedback to end user and this is a problem for sensitive healthcare applications because of high delay. Fog Computing consider to be temporally near to the sensor; thus, will decrease delay [18]. There are several of IoT and OSI application protocols relies on TCP used to communicate and deliver data. This paper, provides an answer to these questions: "Which protocol will be used with low response time and high throughput?", "Which is the best location for servers that represents the lowest delay in order to rapidly send notification to end user" and "Is Fog Computing actually has better performance than Cloud Computing?"

Table 2. Difference between MQTT and HTTP

	MQTT	HTTP
Transport	TCP	TCP
Architecture	Client/Broker	Client/Server
Model	Publish/Subscribe	Request/Response
QoS	3 Types	None
Messages	Topic	URL
Standard	OASIS	Arch. Style
Encoding	Binary	Different Types
Security	Username and Password SSL/TLS	SSL/TLS

4 Literature Review

MQTT and HTTP protocols are used in communication between people and devices especially in the medical field. However, up to our knowledge few papers present the performance of these protocols under conditions such as over large volume of traffic and based on Fog and Cloud layer. The performance testing of XMPP protocol was tested and the evaluation methodology was developed using Tsung traces to check the requirement need of the protocol [19]. Also, the performance of XMPP server was tested using load distributed Tsung over high traffic and from honeypot sensors to find the limit of number of concurrent request [20], but MQTT and HTTP are not included in the above two papers, as well as Cloud and Fog layer are not mention. While in [21], the performance of Web IoT protocols (DDS, XMPP and MQTT) was compared according to the latency of message delivery from sensors and throughput, however these protocols are not implemented in Fog efficiency concept. Among the above works, [22] including Fog and the selection of network management protocols such SNMP, NETCONF and CoAP were evaluated. But, they have mentioned only the management protocols and implemented using OMNeT++ simulator not the real hardware. In [23], MQTT, WebSocket and CoAP application protocols were compared in IoT scenario based on local via Ethernet and remote server via internet and cellular network. However, the response time and throughput of MQTT and HTTP over a huge volume of data were not included. In [24], they proposed system architecture to the problem of middleware, scalability and interoperability between Cloud and sensors. In this system, publisher/subscriber model was applied using MQTT protocol and average response time and throughput was measured. In [25] the overhead and payload size matrices of HTTP and MQTT were compared but without the relation to the Cloud and Fog servers, and then a queuing theory was proposed to evaluate the performance of MQTT.

5 Methodology

The main objective of this paper is comparing the performance of Web IoT protocols with each other, in term of number of sensors that it can handle with low response time and packet loss, and then finding the best location for the servers.The operation of the proposed IoT architecture is as follows: We implement two IoT scenarios as in the Fig. 3and 4, and provide the performance analysis of MQTT and HTTP protocols in six data communication paths: sensors to Fog (located in Al-Nahrain University, College of Information Engineering), sensors to Cloud (located in Ministry of Higher Education and Scientific Research, Department of Research and Development) IaaS/PaaS, sensors to Cloud SaaS (located in different country) and all these steps will be repeated for http protocol. There is a similarity in some of the settings in the two scenarios. Such as, we setup one real pulse sensor1 (heartbeat pulse sensor) and emulate the other sensors using TSUNG (also called Tsunami). With TSUNG, we solve the problem of having hundreds or even thousands of sensors to simulate a real environment — Tsung (also called Tsunami) is an open source program with GPLv2 (General Public License version 2) and developed by Erlang, which provides multi protocols like MQTT and HTTP. For data collection from sensors, we use NodeMCU (also called ESP8266-12E) programmed using C/C++ programming language. Then, these sensors connect to an IEEE802.11n Access Points. The last similar settings consist of two type of server Fog server and Cloud server. So as a whole, APs are connected to Fog layer by using Ethernet, while connected to remote servers via Internet, in both cases with constant bandwidth. There are some different settings in each scenario: in first scenario, the MQTT v3.1.1 protocol with QoS level 0 and 1 is used in the first scenario, MQTT broker (mosquitto) is necessary to mediate the transferring data between subscriber and publisher, then, data is stored using MangoDB5 temporary database with Robomongo GUI through Node.js by using TTL (Time to Live) Fog based. A Path to another MangoDB Cloud based with same configuration and this is a permanent storage and also another path to public (mosquitto) located in different country as shown in Figure 3. As it shown in Figure 4, HTTP protocol v1.1 and GET command are used to request data. The Fog layer is LAMP (Linux, Apache, MySQL, PHP) server used to a temporally store data and Cloud layer is also lamp server but in contrary it considers a permanent storage. The final path is to Dweet Cloud located in different country and Freeboard8 for monitoring data or to infrastructure LAMP Cloud at the same region.

6 Results

This section shows experimental results from the performance analysis with comments. In this paper, one session is programmed in Tsung by using XML v1.0 language and is executed to handle all requests of protocols with this session to do authentication and connection with server side. Also, Tsung is configured to generate a large number of virtual sensors— or what is called the average arrival rate— to publish a huge number of messages only per one physical computer. Finally log level of Tsung is set to type of debug, so that can handle long logging. Also, in order to calculate the response time, throughput and packet loss, every request and message generated by the MQTT and HTTP protocol are recorded using Tsung. At the end, the overall running time of test takes 170 minutes.

Table 3. Size of packet contents (in bytes)

	Message	PDU	Response size
MQTT	75	11	2
HTTP	75	79	67

The performance analysis of the protocols:

6.1 Response Time

The response time of protocols is the elapsed time taken by a web to respond to a request for web services. In this test, the number of sensors set for requesting and publishing data from 100 to 1500 sensors. In Figure 5, architecture based Fog shows that sensors requesting a web page as using HTTP is 12.1 times higher than sensors using MQTT protocol, Cloud based HTTP is 4.7 times higher than MQTT where Cloud located in the same region as the region of sensors, and Cloud based HTTP achieves 2.5 slower than MQTT and the later Cloud located in different coun-

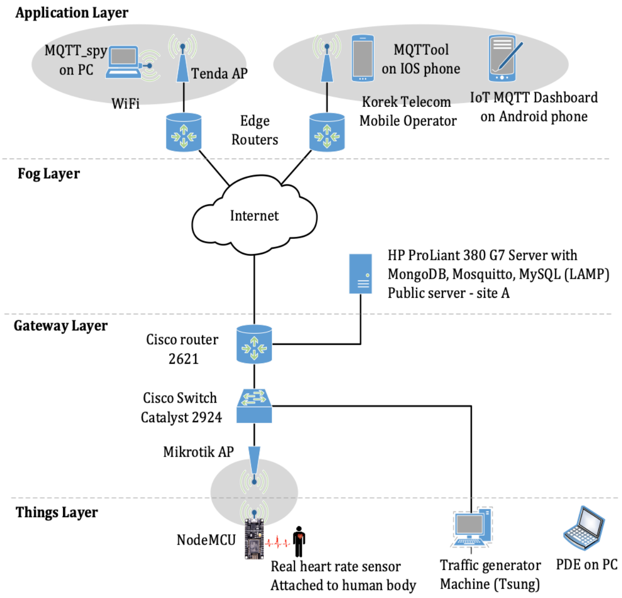

Figure 3. Proposed IoT based Fog architecture

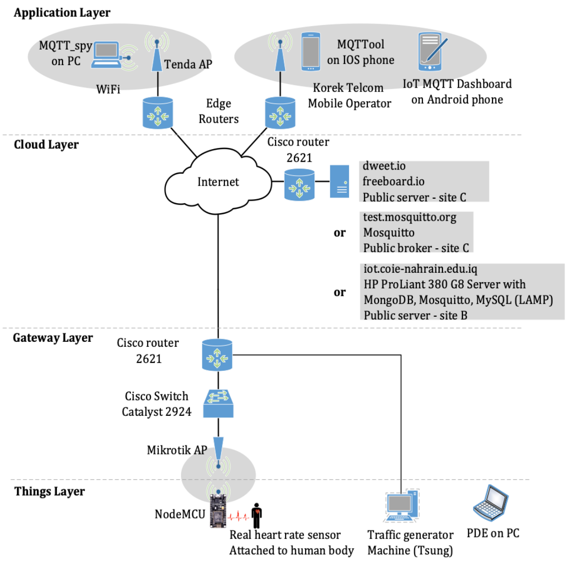

Figure 4. Proposed IoT based Cloud architecture

try, and these results compare with QoS 0 of MQTT, while MQTT QoS level 1 is used with HTTP the results showed: 7.8%, 2.7%, 1.8% as respectively, as an example if number of sensors is 1000. The reason for that is the MQTT has long keep a live time for connection to handle multiple requests and low overhead whereas HTTP opens the TCP connection for short time. Also, the MQTT has low overhead size only 2 Bytes in handshake than HTTP. As a result, it is not efficient that sensors depend on Cloud for processing data and send feedback to interested persons.

Table 4. Performance between sensors and Fog/Cloud

Metric	Type of Server	Bandwidth	Protocol
Response Time	Cloud	20.4 Mbits/sec	HTTP
	Fog	89.3 Mbits/sec	HTTP
	Cloud	26.8 Mbits/sec	MQTT QoS 0
	Cloud	26.8 Mbits/sec	MQTT QoS 1
	Fog	93.9 Mbits/sec	MQTT QoS 0
	Fog	94.0 Mbits/sec	MQTT QoS 1
Throughput and Packet Loss	Cloud	4.11 Mbits/sec	HTTP
	Fog	6.05 Mbits/sec	HTTP
	Cloud	6.53 Mbits/sec	MQTT QoS 0
	Cloud	16.4 Mbits/sec	MQTT QoS 1
	Fog	5.72 Mbits/sec	MQTT QoS 0
	Fog	7.64 Mbits/sec	MQTT QoS 1

6.2 Throughput

Throughput is the amount of data that server could handle in period of time. The next Figure 6, shows that the throughput of two protocols MQTT and HTTP. Throughput performance shows that HTTP is 7.1 times higher than MQTT protocol QoS 0 Fog based or 6.38 times higher in Cloud based (located in the same region). The location of the server does not impact so much on throughput performance of both protocols and even if it impacted, factor 1 will be affected. Also, we notice that the HTTP protocol has reached the saturation level earlier than MQTT protocol in both cases Fog or Cloud based. The impact performance of throughput depends on server capabilities to handle data and load.

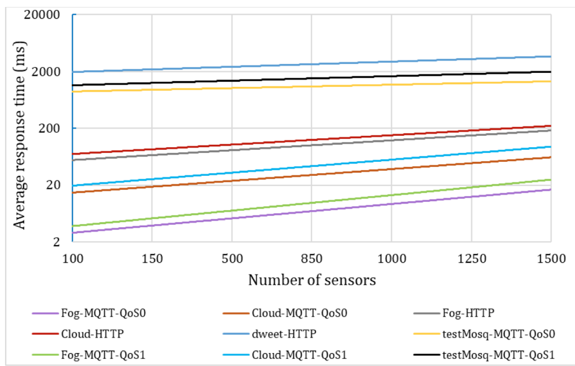

Figure 5. Average response time of proposed IoT based Fog and Cloud architectures

6.3 Packet Loss

The packet loss defined as number of packets of data fail to reach the final destination when they travel through network. In Figure 7 below, packets loss was compared in terms of the number of messages that

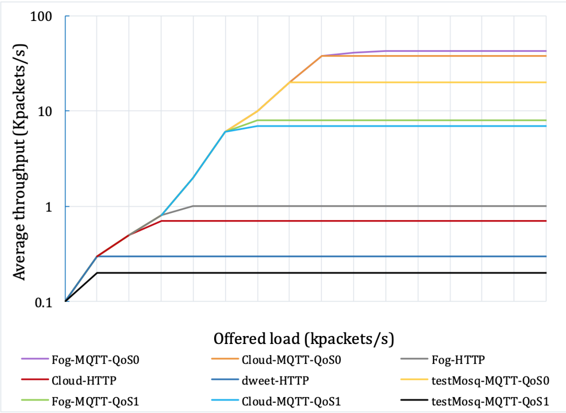

Figure 6. Average throughput of proposed IoT based Fog and Cloud architectures

published and requested, the two protocols MQTT and HTTP, and QoS levels and location of local and remote servers. The results shows MQTT QoS 1 packet loss is 6.2 times higher than MQTT QoS 0 Fog based. Also, HTTP message loss is 49.7 higher than MQTT QoS 0 Fog based and in case of Cloud based located in the same region it would be 41.1 higher than MQTT QoS 0, as an example if the number of messages per sec is 50,000. And all this happened because of the following reasons: MQTT has lowest handshake and lowest PDU, Fog has a lower packet loss than Cloud because the Fog is local and there is no need for the network to have routers, these routers are unable to hold traffic with limited bandwith, unlike Cloud. In addition, the path to the Cloud may contain multiple routers connecting together by links, if one of these links is busy the packets have to wait in the queue. Also, if the queue is at full capacity the packets will be dropped. Furthermore, the packet loss impact on response time of protocols because of the retransmission of lost packets, thus leads to higher response time.

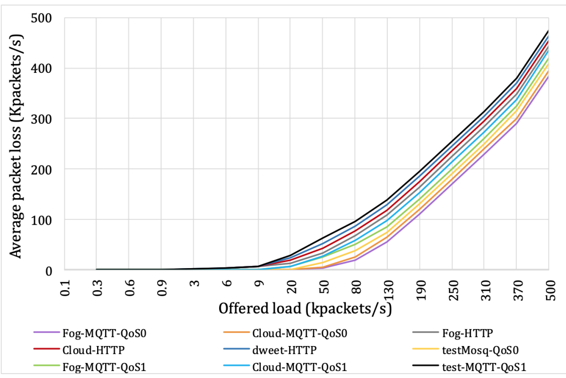

Figure 7. Average packet loss of proposed IoT based Fog and Cloud architectures

7 Conclusion

In this paper, the proposed IoT architecture suggest a middle layer named Fog consists of high speed temporary storage to enable fast end users reporting. We perform an experimental setup to analyze two Web IoT protocols: MQTT and HTTP. We implement these protocols using low cost embedded devices with private and public servers working as Fog and Cloud (there are two Cloud: one in the same region with Fog and other at different country). The work concentrates on generating large traffic volume from sensor-like terminal running Tsung tool integrated with real heart sensor traffic to simulate the required scenarios. The obtained results of response time and throughput for both scenarios (Fog based and Cloud based) show that the MQTT protocol advances the HTTP protocol since the latte one consists of an extra handshaking and more overhead than MQTT. On the other hand, using Fog servers as a middleware layer close to the embedded devices (organization levels) enhances performance and this is clearly shown in the results obtained. Fog servers may be designed as close as possible to the end user devices in distributed layers. While, the throughput in both scenarios is related directly to the available bandwidth between the gateway and Fog/Cloud servers.

References

[1] Ferrag, M. A., Derdour, M., Mukherjee, M., Derhab, A., Maglaras, L., & Janicke, H. (2018). Blockchain Technologies for the Internet of Things: Research Issues and Challenges. IEEE Internet of Things Journal, 1-1. doi:10.1109/jiot.2018.2882794

[2] Al-Joboury, I. M., & Al-Hemiary, E. H. (2017). Internet of Things (IoT): Readme. Qalaai Zanist Scientific Journal, 2(2), 343-358. doi:10.25212/lfu.qzj.2.2.35

[3] Krasniqi, X. (2016). Internet of Things: From applications, challenges and standardization to Industry implementations. International Journal of Business & Technology, 4(2). doi:10.33107/ijbte.2016.4.2.03

[4] Shariatmadari, H., Iraji, S., & Jäntti, R. (2017). From Machine-to-Machine Communications to Internet of Things: Enabling Communication Technologies. From Internet of Things to Smart Cities, 3-34. doi:10.1201/9781315154503-1

[5] Al-Joboury, I. M., & Hemiary, E. H. (2018). Internet of Things Architecture Based Cloud for Healthcare. Iraqi Journal of Information & Communications Technology, 1(1), 18-26. doi:10.31987/ijict.1.1.7

[6] Mishra, D. P., Kumar, S., & Ashu, A. (2018). Internet of Things: A Survey on Enabling Tech-

nologies, Application and Standardization. SSRN Electronic Journal. doi:10.2139/ssrn.3170749

[7] Stojmenovic, I., & Wen, S. (2014). The Fog Computing Paradigm: Scenarios and Security Issues. Proceedings of the 2014 Federated Conference on Computer Science and Information Systems. doi:10.15439/2014f503

[8] Mahmud, R., Kotagiri, R., & Buyya, R. (2017). Fog Computing: A Taxonomy, Survey and Future Directions. Internet of Things Internet of Everything, 103-130. doi:10.1007/978-981-10-5861-5_5

[9] Al-Joboury, I., & Al-Hemiary, E. (2018). IoT-F2CDM-LB: IoT Based Fog-to-Cloud and Data-in-Motion Architectures with Load Balancing. EAI Endorsed Transactions on Internet of Things, 4(13), 155332. doi:10.4108/eai.6-4-2018.155332

[10] Aazam, M., & Huh, E. (2014). Fog Computing and Smart Gateway Based Communication for Cloud of Things. 2014 International Conference on Future Internet of Things and Cloud. doi:10.1109/ficloud.2014.83

[11] Lee, S., Kim, H., Hong, D., & Ju, H. (2013). Correlation analysis of MQTT loss and delay according to QoS level. The International Conference on Information Networking 2013 (ICOIN). doi:10.1109/icoin.2013.6496715

[12] Kim, G., Park, J., & Chung, K. (2018). Priority-based Multi-level MQTT System to Provide Differentiated IoT Services. Journal of KIISE, 45(9), 969-974. doi:10.5626/jok.2018.45.9.969

[13] Masek, P., Hosek, J., Zeman, K., Stusek, M., Kovac, D., Cika, P., . . . Kröpfl, F. (2016). Implementation of True IoT Vision: Survey on Enabling Protocols and Hands-On Experience. International Journal of Distributed Sensor Networks, 12(4), 8160282. doi:10.1155/2016/8160282

[14] Al-Joboury, I. M., & Al-Hemiary, E. H. (2017). F2CDM: Internet of Things for Healthcare Network Based Fog-to-Cloud and Data-in-Motion Using MQTT Protocol. In: Sabir E., García Armada A., Ghogho M., Debbah M. (eds) Ubiquitous Networking. UNet 2017. Lecture Notes in Computer Science, 368-379. doi:10.1007/978-3-319-68179-5_32

[15] Information technology. Message Queuing Telemetry Transport (MQTT) v3.1.1. (n.d.). doi:10.3403/30328634

[16] Belshe, M., & Peon, R. (2015). Hypertext Transfer Protocol Version 2 (HTTP/2). doi:10.17487/rfc7540

[17] Alrawais, A., Alhothaily, A., Hu, C., & Cheng, X. (2017). Fog Computing for the Internet of Things: Security and Privacy Issues. IEEE Internet Computing, 21(2), 34-42. doi:10.1109/mic.2017.37

[18] Dastjerdi, A. V., & Buyya, R. (2016). Fog Computing: Helping the Internet of Things Realize Its Potential. Computer, 49(8), 112-116. doi:10.1109/mc.2016.245

[19] Che, X., & Maag, S. (2014). Testing protocols in Internet of Things by a formal passive technique. Science China Information Sciences, 57(3), 1-13. doi:10.1007/s11432-014-5068-x

[20] Kurniawan, K. J. , Lim, C. and Eng, K. I, (2015). XMPP Performance Analysis using large volume traffic from honeypot sensor. The International Conference on Innovation, Enterpreneurship, and Technology (ICONIET 2015).

[21] Babovic, Z. B., Protic, J., & Milutinovic, V. (2016). Web Performance Evaluation for Internet of Things Applications. IEEE Access, 4, 6974-6992. doi:10.1109/access.2016.2615181

[22] Slabicki, M., & Grochla, K. (2016). Performance evaluation of CoAP, SNMP and NETCONF protocols in fog computing architecture. NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium. doi:10.1109/noms.2016.7503010

[23] Mijovic, S., Shehu, E., & Buratti, C. (2016). Comparing application layer protocols for the Internet of Things via experimentation. 2016 IEEE 2nd International Forum on Research and Technologies for Society and Industry Leveraging a Better Tomorrow (RTSI). doi:10.1109/rtsi.2016.7740559

[24] Triawan, M. A., Hindersah, H., Yolanda, D., & Hadiatna, F. (2016). Internet of things using publish and subscribe method cloud-based application to NFT-based hydroponic system. 2016 6th International Conference on System Engineering and Technology (ICSET). doi:10.1109/icsengt.2016.7849631

[25] Yokotani, T., & Sasaki, Y. (2016). Comparison with HTTP and MQTT on required network resources for IoT. 2016 International Conference on Control, Electronics, Renewable Energy and Communications (ICCEREC). doi:10.1109/iccerec.2016.78149

Istabriq M. Aljoboury Istabriq M. Aljobouri has B.Sc. in 2015, M.Sc. in 2017 from Department of Networks Engineering, College of Information Engineering at Al-Nahrain University Bagdad – Iraq. She is specialized in Internet of Things branch of research where she published some articles as a part of her M.Sc. project. She is currently applying for completing Ph.D. in that field.

Emad H. Al-Hemiary Emad Al-Hemiary graduated from the College of Engineering at Al-nahrain University and have his B.Sc. (1993), M.Sc. (1996), and Ph.D. (2001) under the specialization of Electronics and Communications Engineering/Electronics Systems Engineering and Network Design. He participated in the establishment of the College of Information Engineering at Al-Nahrain University and worked as a faculty member there where he published many articles in related fields of Interests.