Impact of Link Delay Variation on MQTT and CoAP based Communication Performances in Mobile Environment

OUAKASSE Fathia*† and RAKRAK Said*

*Applied Mathematics and Computer Science Laboratory (LAMAI)
Cadi Ayyad University, Marrakesh,Morocco.

Summary
In a mobile network, the great challenge in the Internet of Things IoT is the choice of which application layer protocol to use for lightweight devices and constrained resources like Wireless Sensors Network WSN. Moreover, these constrained devices communicate very frequently using a large amount of messages and notifications which cause the handover problem. In this sense, two of the most emerging messaging protocols appropriate to address the needs of these lightweight IoT nodes are Message Queuing Telemetry Transport (MQTT) and Constrained Application Protocol (CoAP). In this paper, we present a description of these two protocols then we evaluate their performances in the context of mobile or dynamic networks. In addition, in order to evaluate the behavior of MQTT and CoAP protocols in such a network under the condition of a mobile environment, we draw emulations based on different link delay parameters.

Key words:
MQTT, CoAP, Link Delay, Packet Loss, Performance, Mobile Environment.

1. Introduction

Recently, Wireless sensor networks (WSNs) have been widely used and are deployed in many applications in order to measure, control or detect physical and environmental events like pressure, humidity, temperature and pollution levels, as well as other critical parameters. Usually, applications send queries to concerned sensors to retrieve values periodically from the measurements or detections. However, in recent critical applications of WSN which require intervention, such as home automation, industry process control, healthcare, environment monitoring, smart grid and ambient assisted living, the challenge is getting information when an event of interest occurs in order to intervene in real-time. In this context, the publish/subscribe model [1] is presented as the most appropriate model covering these requirements. This model includes two essential entities; the subscriber and

the publisher. Subscribers are the entities that express their interest in events produced by publishers. They can carry out a subscription in different events. While, publishers are entities that generate information in order to be forwarded to the interested subscribers. This model presents many advantages like the asynchronous way in which this interaction is performed and the fact that subscribers and publishers can exchange information without neither the need to know about each other nor the need of being actively participating in the interaction at the same time. These features make the pub/sub model more scalable and flexible. That's the reason why it is highly suitable for WSN and for current trends of IoT.

Two of the most important protocols based on this model are MQTT [2] and CoAP [3]. This goes back to the fact that they are the most appropriate protocol for lightweight devices and constrained resources in terms of memory, energy, and computing.

However, these protocols have not been thoroughly tested and evaluated in the context of mobile or dynamic networks while mobility management is very crucial in such networks using such devices. Thus, in this paper, we present an evaluation of these protocols under different scenarios by emulations based on Core network emulator. We try to characterize their behavior in terms of throughput, latency, packet loss and jitter values. Based on the results obtained, we provide criteria of applicability of these protocols, and we assess their performance and viability based on the link delay variation. This evaluation is of interest for the upcoming systems applications related to the Internet of Things.

The remainder of this paper is organized as follows: a description of the two protocols considered in this evaluation MQTT and CoAP is presented as background in the second section. Then, in the third section related works to mobility management are discussed. Afterward, the performances of MQTT and CoAP protocols in the case of mobility based on different link delay parameters are evaluated using the Core network emulator and based on the results obtained a discussion is opened in the fourth. Finally, in the fifth section, conclusion and some perspectives are closing up our paper.

Manuscript received August 5, 2017
Manuscript revised August 20, 2017

2. Background

2.1 MQTT overview

MQTT is a lightweight messaging protocol oriented to be used in resource-constrained devices and machine-to-machine (M2M) interactions in the mobile sector, it was proposed by OASIS to support IoT communications. MQTT is an application layer; it includes three components a subscriber, a publisher and a broker and uses a topic-based publish-subscribe architecture as shown in Figure 1.

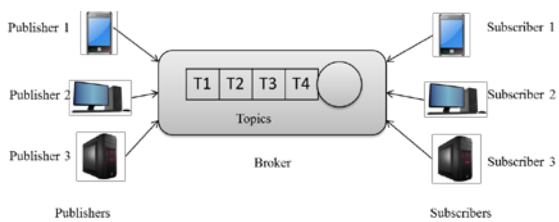


Fig .1. MQTT protocol architecture

The communication between the subscribers and the publishers is performed in three steps:

- Subscribers subscribe to an or many particular topics in which they are interested.
- Clients or publishers start publishing messages on different topics in the broker.
- All subscribers subscribed to these topics receive the messages published via the broker.

The process of the communication between the three components of MQTT is drawn Figure 2.

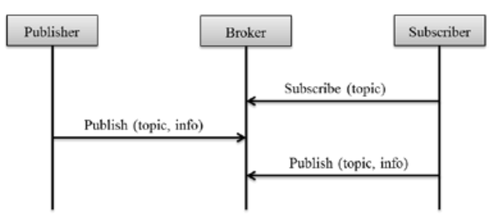


Fig .2. The synchronization between the three components of MQTT

Nowadays, two main specifications exist for MQTT: (i) MQTT v.3.1 [4] and (ii) MQTT-SN (also known as MQTT-S) [5].

MQTT protocol provides a support for delivering messages between publishers and subscribers. QoS is an attribute of an individual MQTT message being published. While, for the QoS of a message forwarded to a subscriber, it isn't necessary to be the same as the QoS given to the message by the original publisher.

Indeed, a mechanism of acknowledgments exchange is taken place between the client and the broker in order to ensure that messages have been received. This mechanism is associated with a quality of service level specified on each message [6].

- QoS level zero (QoS=O): the sender sends the message only once and no retries are performed; fire and forget. Messages sent might be lost. This level is depicted in Figure 3a.

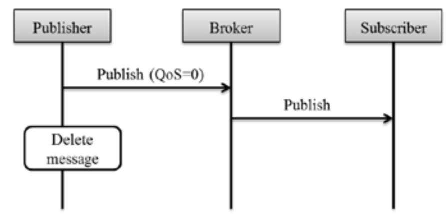


Fig.3a. MQTT QoS = 0: At most one

- QoS level one (QoS=1): to ensure that the message arrives at its destination at least once, the published message is stored in the publisher internal buffer until it receives the ACK packet. Once the acknowledgment is received, the message is deleted from the buffer. This level is drawn in Figure 3b.

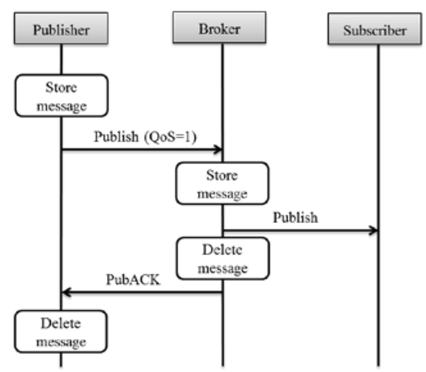


Fig.3b. MQTT QoS = 1: At least once

- QoS level two (QoS=2): the protocol guarantees that a published message will be delivered exactly once. In this level two-step acknowledgment process, as drawn in Figure 3c, is used in order to assure that neither loss nor duplication of messages will happen.

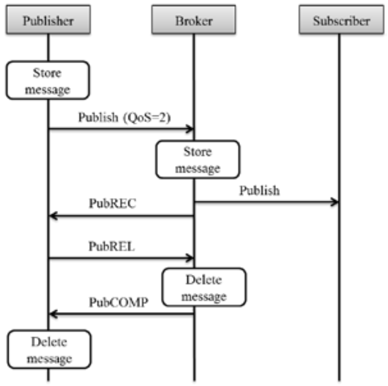


Fig.3c. MQTT QoS = 2: Exactly once

2.2 CoAP overview

CoAP has been designed by the Internet Engineering Task Force (IETF) to support IoT with lightweight messaging for devices operating in a constrained environment.

CoAP is an application layer protocol based on a REST architecture. It defines two kinds of interactions between end-points:

- The client/server model which provides as well two interaction types:
  - ✓ A one-to-one interaction: request/reply.
  - ✓ A multi-cast interaction: from a server to multiple clients. Clients have the ability to manage resources using requests: GET, PUT, POST and DELETE to perform Create, Retrieve, Update, and Delete operations.
- The publish/subscribe model called the observer model [7], where a server, playing the role of the publisher, sends messages of notifications as publications to an observer, playing the role of subscriber, about a resource (event) that the subscriber is interested in receiving.

Indeed, CoAP runs over UDP; UDP broadcasts and multicasts are allowed by CoAP for addressing [8]. Thus, CoAP is considered more suitable for the IoT domain, this is going back to the fact that it is possible to build sufficiently basic error checking and verification for UDP to make sure that messages arrive without the significant communication overhead as in the case of TCP [9]. An overview architecture of CoAP protocol is drawn in Figure 4.

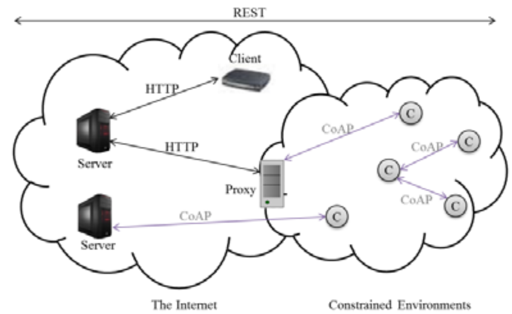


Fig. 4. An overview architecture of CoAP protocol

Like MQTT, in order to ensure that messages have been received, CoAP defines two types of QoS called reliability, it defines a confirmable message and a non-confirmable message [10]. In the case of a confirmable message an acknowledgment message (ACK) is sent to the sender from the intended recipient as shown in Figure 5a, else the message is retransmitted.

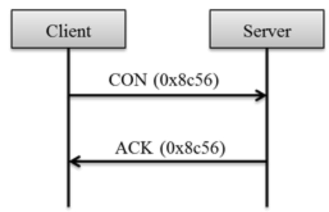


Fig. 5a. Reliable message transport

However, in the case of a non-confirmable message, no reception confirmation is expected as shown in Figure 5b [11].

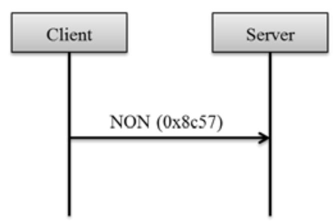


Fig. 5b. Unreliable message transport

2.3 Mobility management

MQTT is a many-to-many communication protocol providing messaging interaction between multiple subscribers and multiple publishers through a central broker. After the subscriber's subscriptions and the publisher's publications, MQTT let the broker decide where to route and copy messages according to the topic's subscriptions.

However, this protocol presents some limitations especially in mobile environments and as we know, it is sure that the efficient handling of mobility is crucial for the overall performance of IoT applications. In this context and to overcome this problem, authors in [12] propose to store messages in the buffer placed on each access point during the handover of the mobile terminal. At the end of the handover, the current access point transfers messages to the new access point. In the same context, a seamless handover for a hotspot network using a buffering technique is proposed in [13] as an intermediate buffering and it was evaluate in various scenarios where the publisher node suffers a handover process due to the mobility. Thus when the connection between the publishers and the broker undergoes an interruption, nodes enter in roam mode and messages that haven't been acknowledged from the broker are stored in the MQTT internal buffer which has a limited capacity. Only after recovering the connection with the last access point (last IP address) that these messages are delivered, unless they are lost. So, the intermediate buffer stores non acknowledged messages in case of a longer connection interruption. A basic diagram of this mechanism is drawn in Figure 6.

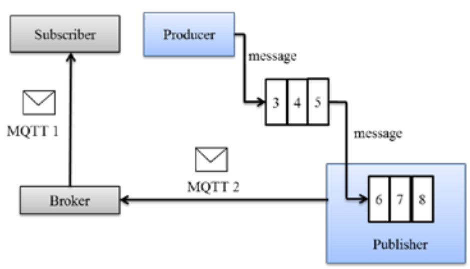


Fig. 6. Structure of the mechanism based on intermediate buffering

The same work was extended in [14] where authors add a cross-layer solution in addition to the intermediate buffer located on each mobile terminal. This improves the device connectivity according to the data layer management and guarantees that no information is lost in the data delivery.

On the other hand, CoAP has become increasingly proposed for gathering data from smart sensors and controlling constrained devices. However, as MQTT, the mobility of device represents the great limitation that hinders the proper functioning of CoAP protocol and causes the loss of packets. To prevent the mobility problems, there are many solutions in the literature developed by the IETF like the mobility management for network layer, Mobile IPv4/v6 (MIPv4/v6) [15] and its variants, including Fast Mobile IPv4/v6 (FMIPv4/v6) [16], Hierarchical Mobile IPv4/v6 (HMIPv4/v6) [17], and Proxy Mobile IPv4/v6 (PMIPv4/v6) [18].

Moreover, based on the specification of the Proxy Mobile IPv6 (PMIPv6) [18], authors in [19] propose two network-based mobility management schemes CoAP-PMIP and CoAP-DPMIP. In CoAP-PMIP, they define a Local Mobility Anchor (LMA) to store the address information of mobile sensors. However, in CoAP-DPMIP, a mechanism is proposed in order to provide a more optimized data transmission path and also to reduce the handover delay.

Furthermore, in [20] a mobile CoAP for mobility management (CoMP) is presented where authors propose to keep track of the current IP addresses of the mobile sensor during the handover and using both HTTP and CoAP, enable sensed data to be reliably delivered to the Web clients.

3. Link delay impact in MQTT and CoAP communication in case of mobile devices

MQTT and CoAP protocols are being implemented for mesh-networking applications in networks, in order to allow inter-standard communication between lightweight end nodes. However, the use of which of each of these protocols depends on the scenario and the experiment conditions. In this section, we present the emulation results of MQTT and CoAP based communication scenarios using different link delays under the condition of mobile devices.

Indeed, to our knowledge, this is the only work that test, evaluate and compare the behavior of each of these protocols in a mobile network. Next to this, there exists several works in which MQTT and CoAP protocols are together evaluated like in [21] where authors present a comparative study of these two protocols via a real experimentation. In [22] authors present an evaluation of the performance of MQTT and CoAP via a Common Middleware. However, a study of industrial protocols in the IoT including MQTT and CoAP is presented in [23, 24]. Furthermore, in [25] authors propose a combined exploitation of MQTT and CoAP in order to achieve better scalability.

In order to evaluate the performances of MQTT and CoAP using different link delay parameters under the scenario of mobile devices, we perform emulations. Using the Core network emulator, we draw two scenarios in a mobile environment based on ENAME model mobility: (i) fixed packet loss to 1% and variable link delay and (ii) fixed link delay to 10ms and variable packet loss. The parameters considered in this emulation are detailed in Table1.

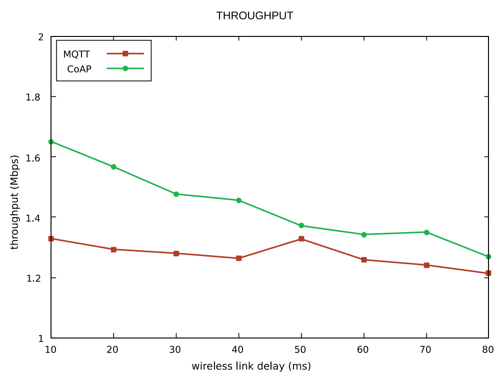
Table 1: Emulation parameters

Parameters	Values
Data traffic	100 messages
Message size	6 ko, 8ko
Link packet loss	1%, 30%
Link delay	10 ms, 80 ms
Duration	0 sec, 100 sec

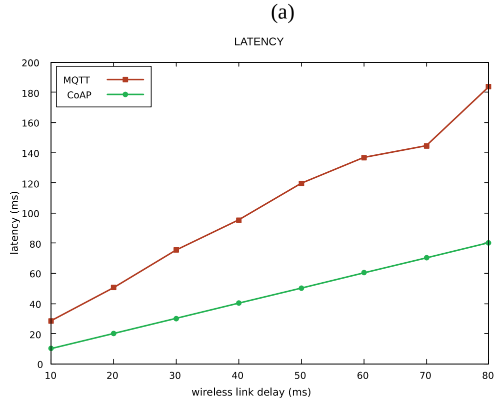
3.1 Case of fixed link packet loss

Figures 7a, b and c show the results of throughput, latency and jitter using the scenario of a fixed link packet loss and variable link delay. Results show that CoAP performs well in terms of throughput, latency and presents a higher jitter compared to MQTT. These results are explained with the fact that MQTT and CoAP use different transmission protocols of TCP and UDP.

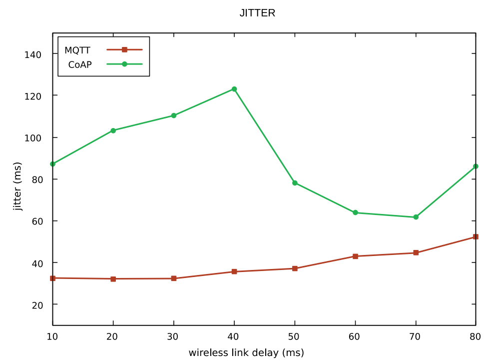
Furthermore, the packet loss presented by each protocol according to link delay variation is presented in Figure 7d. Initially, in low link delay, MQTT and CoAP have slightly the same average packet loss. Afterward, graphs start to increase for MQTT and decrease for CoAP, this is due to the fact that congestion is likely to happen in high link delay causing more retransmissions delays and losses of packet.



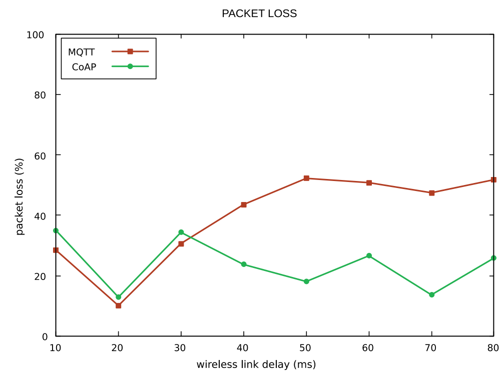
(a)



(b)



(c)



(d)

Fig.7. Throughput (a), Latency (b), Jitter (c) and Packet loss ratio (d) according to Wireless link delay variation using a fixed link packet loss

3.2 Case of fixed link delay

As expected, using a fixed link delay, CoAP still performs in terms of throughput, however, as the link packet loss increases; CoAP performance decreases until its graph meet the MQTT graph in a low throughput value. Indeed, the size of messages and the frequency in which they are sent generate an important error ratio either as losses or as congested messages, thing which make a great impact in MQTT and CoAP throughput values as shown in Figure 8a. Nevertheless, in terms of latency (Figure 8b), the graphs are reversed; the same performances are presented by MQTT and CoAP in low link packet loss, but MQTT latency has dramatically increased with the increase of link packet loss. On the other hand, in terms of jitter and packet loss, MQTT and CoAP have slightly the same curve variation as drawn in Figure 8c and d.

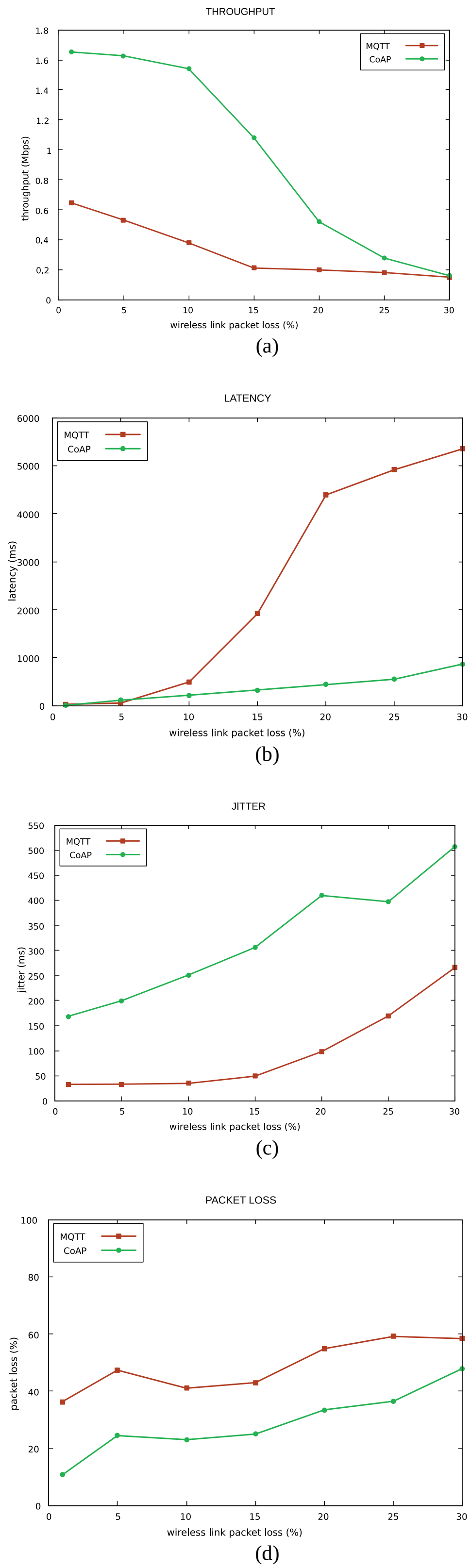


Fig.8. Throughput (a), Latency (b), Jitter (c) and Packet loss ratio (d) according to Wireless link packet loss variation using a fixed link delay

We can say that the link delay has an important impact in the performances of MQTT and CoAP. The purpose of this evaluation is to choose the suitable link delay for a MQTT or CoAP based communication in a mobile environment; this is according to the best throughput and lowest latency resulted in the presence of different metrics (link packet loss and data traffic offered by the devices).

So, in the choice of the most suitable messaging solution for an application in a mobile environment, the study of the choice should not only be based on an understanding of the architecture but also the main application requirements in terms of link delay and other performances parameters as emulation results have proved.

4. Conclusion

In many critical application fields like industry process and health, the mobility of devices must be managed to ensure the reliable data transmission. Thus in order to achieve mobility management, authors design architectures and mechanisms to support mobility management while considering the characteristics of the constrained devices. The challenge, in such networks i.e. constrained devices in a distributed mobile network, is each of which application protocol to use. The reason why, in this paper, we conduct a description of two of the most appropriate protocol to address the lightweight devices and we evaluate their performances according to the link delay variation then we provide criteria of applicability of these protocols, in order to assist programmers in their choice decision process.

References

[1] Eugster PT, Felber PA, Guerraoui R, Kermarrec AM. The many faces of publish/subscribe. ACM Computing Survey. June 2003; 35. pp. 114–131.

[2] MQTT.org, Mq telemetry transport. http://mqtt.org/, (2013, accessed 05/05/2017).

[3] Shelby Z, Hartke H, Bormann C. Constrained application protocol (coap) draft-ietf-core-coap-17. http://tools.ietf.org/html/draft-ietf-core-coap-17, (2013, accessed 05/05/2017).

[4] Locke D. MQ Telemetry Transport (MQTT): V3.1 Protocol Specification. IBM: Developer Works: http://goo.gl/3tLZVj. (August 2010 08/06/2017).

[5] Stanford-Clark A, Truong H L. MQTT For Sensor Networks (MQTT-SN): Protocol Specification Version 1.2. http://goo.gl/eDqIRQ. (November 2013 accessed 01/07/2017).

[6] Patierno P. Smart home and smart factory systems: MQTT and IoT protocols comparison. http://fr.slideshare.net/paolopat/mqtt-iot-protocols-comparison?next_slideshow=1 (15 February 2014, accessed 09/07/2017).

[7] Hartke K. Observing Resources in CoAP Draft-Ietf-Core-Observe-06," RFC 7641, Ver. 06, 2012.
[8] Jaffey T. MQTT and CoAP, IoT Protocols. https://eclipse.org/community/eclipse_newsletter/2014/february/article2.php. (2014, accessed 25/07/2017).
[9] Masek P, Hosek J, Zeman K, Kröpfl F. Implementation of True IoT Vision: Survey on Enabling Protocols and Hands-On Experience. International Journal of Distributed Sensor Networks, Article ID 8160282, 2016, pp. 1-18.
[10] Davis EG, Calveras A, Demirkol I. Improving Packet Delivery Performance of Publish/Subscribe Protocols in Wireless Sensor Networks. Journal of sensors, vol. 13, 2013, pp. 648-680.
[11] Chen X. Constrained Application Protocol for Internet of Things. http://www.cse.wustl.edu/~jain/cse574-14/ftp/coap/index.html. 2014. (Accessed 06/06/2017).
[12] Yamagata F, Komaba D, Oguma H, Kameda S, Nakase H, Takagi T, Tsubouchi K. Seamless handover for hotspot network using buffered packet forwarding method. In IEEE Singapore International Conference on Communications Systems (ICCS '06), Singapore, Nov. 2006, pp. 1 - 5.
[13] Luzuriaga JE, Cano JC, Calafate C, Manzoni P, Perez M, Boronat P. Handling Mobility in loT applications using the MQTT protocol. In IEEE conference on Internet Technologies and Applications (ITA), Wrexham, North Wales, UK, 8-11 Sept. 2015, pp. 245 - 250.
[14] Luzuriaga JE, Perez M, Boronat P, Cano JC, Calafate C, Manzoni P. Improving MQTT Data Delivery in Mobile Scenarios: Results from a Realistic Testbed. Mobile Information Systems J 2016, Article ID 4015625, 11.
[15] Johnson D, Perkins C, Arkko J. Mobility support in IPv6. June 2004. Available online: http://www.rfc-editor.org/info/rfc3775, (accessed on 23/06/2017).
[16] Koodli R. Fast handovers for mobile IPv6. July 2005. Available online: http://tools.ietf.org/html/rfc4068 (accessed 05/07/2017).
[17] Soliman H, Castelluccia C, El Malki K, Bellier L. Hierarchical mobile IPv6 mobility management. August 2005. Available online: https://tools.ietf.org/html/rfc4140 (accessed 07/07/2017).
[18] Gundavelli S, Leung K, Devarapalli V, Chowdhury K, Patil B. Proxy Mobile IPv6. Available online: http://www.rfc-editor.org/info/rfc5213 (August 2008, accessed 03/07/ 2017).
[19] Choi SI, Koh SJ. Use of Proxy Mobile IPv6 for Mobility Management in CoAP-Based Internet-of-Things Networks. In IEEE Communications Letters, Vol.20, Issue 11, 18 August 2016, pp. 2284 – 2287.
[20] Chan SM, Park JT. Mobile CoAP for IoT mobility management. 12th Annual IEEE Consumer Communications and Networking Conference (CCNC). 9-12 Jan. 2015. pp. 283 – 289.
[21] Mijovic S, Shehu E, Buratti C. Comparing Application Layer Protocols for the Internet of Things via Experimentation. In IEEE 2nd International Forum on Research and Technologies for Society and Industry Leveraging a better tomorrow (RTSI), Bologna, Italy, 7-9 Sept. 2016, pp.1-5.
[22] Dinesh T, Xiaoping M, Alvin V, Hwee-Xian T, Colin Keng-Yan T. Performance Evaluation of MQTT and CoAP via a Common Middleware. In IEEE Ninth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), Singapore, 21-24 April 2014, pp.1-6.
[23] Foster A. Messaging Technologies for the Industrial Internet and the Internet of Things: A Comparison between DDS, AMQP, MQTT, JMS, REST and CoAP, version 1.2, November 2013.
[24] Foster A. Messaging Technologies for the Industrial Internet and the Internet of Things Whitepaper: A Comparison between DDS, AMQP, MQTT, JMS, REST and CoAP, version 1.7, July 2014.
[25] Bellavista P and Zanni A. Towards Better Scalability for IoT-Cloud Interactions via Combined Exploitation of MQTT and CoAP. In IEEE 2nd International Forum on Research and Technologies for Society and Industry Leveraging a better tomorrow (RTSI), Bologna, Italy, 7-9 Sept. 2016, pp.1-6.

Fathia OUAKASSE MS degree in Telecommunications and Computer Sciences from the National Institute of Posts and Telecommunications, Rabat (Morocco) in 2011. PhD student in the Laboratory of Applied Mathematics and Computer Science, Faculty of Science and Techniques, Cadi Ayyad University, Marrakesh, Morocco.

Said RAKRAK Professor and researcher in the Laboratory of Applied Mathematics and Computer Science, Faculty of Science and Techniques, Cadi Ayyad University, Marrakesh, Morocco.