Journal of Engineering and Applied Sciences 13 (15): 6053-6061, 2018
ISSN: 1816-949X

A Comparative Study of MQTT and CoAP Application Layer Protocols via. Performances Evaluation

Fathia Ouakasse and Said Rakrak
Applied Mathematics and Computer Science Laboratory (LAMAI),
Cadi Ayyad University, Marrakesh, Morocco

Abstract: Everyday objects are generating a large amount of data via. different technologies like Radio Frequency Identification (RFID) and Wireless Sensors Network (WSN). This data should be stored and communicated via. protocols. Two of the most important protocols used in the Internet of Things for lightweight devices and constrained resources and based on publish/subscribe model are Message Queuing Telemetry Transport (MQTT) and Constrained Application Protocol (CoAP). In this study, we describe and compare these two emerging messaging protocols to address the needs of these lightweight IoT nodes, we reveal some of their limitations then we quantitatively evaluate their performances using a network emulator allowing to emulate throughput, latency and inter-arrival time using different scenarios and metrics and finally, we conclude with some future directions.

Key words: Application layer protocols, CoAP, comparison, MQTT, performances, scenarios

INTRODUCTION

Nowadays, objects have the ability to generate a large amount of data thus, it should communicate with each other or with a corresponding node to transfer data via. the internet. This task is accomplished almost always thanks to the use of WSN.

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. Applications usually used to 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 react in real-time. In this context, the publish/subscribe model (Eugster *et al*., 2003) is the most appropriate model covering these requirements. This model includes two fundamental entities, the first entity is called the subscriber and the second one is called the publisher. Subscribers have the ability to express their interest in events produced by publishers. The registration of subscribers in different events is called subscription. On the other hand, publishers are entities that generate information in order to be forwarded to the interested subscribers. This model allows subscribers to receive information without the need of being actively participating in the interaction with publishers at the same time. Moreover, in this model publishers and subscribers do not need to know about each other.

Furthermore, this interaction is performed in an asynchronous way that means that publishers and subscribers can publish or receive event information in a different time slot. In other words, subscribers can receive information even if they perform an activity and publishers as well are not blocked while producing events. This feature makes the pub/sub model more scalable and flexible and provides more dynamic network topology. That's the reason why it is highly suitable for WSN and for current trends of IoT.

The most important protocols based on this model are MQTT, CoAP, AMQP, XMPP and DDS but only two protocols are highlighted in this study, MQTT (Anonymous, 2014) and CoAP. This goes back to the fact that MQTT and CoAP are the most used and appropriate protocol for lightweight devices and constrained resources in terms of memory, energy and computing. So, in the following of this study, we will describe the different aspects of these two protocols as well as a comparative study of their essential characteristics is drawn.

Corresponding Author: Ouakasse Fathia, Department of Computer Science, Cadi Ayyad University, Marrakesh, Morocco

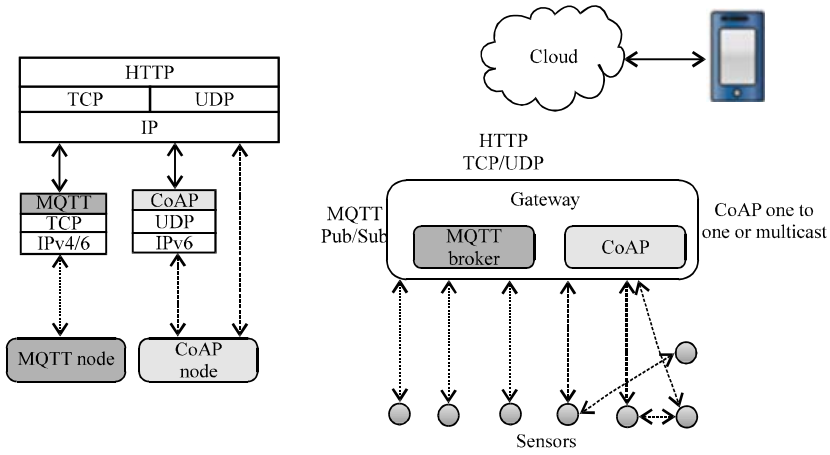

Fig. 1: MQTT and CoAP overview architectures (Stransberry, 2015)

Recently, these two protocols have been widely used in different application fields, we mention some applications, for example, MQTT has been used for remote monitoring applications (Robinson *et al*., 2005; Maynard, 2011), messaging applications (Zhang, 2011) and a range of home automation applications (Shin *et al*., 2015). Furthermore, MQTT has been used also for education control (Prada *et al*., 2016).

On the other hand, CoAP has been deployed in several application fields for resource constrained networks and M2M applications ranging from smart grid, building and home automation (Bergmann *et al*., 2012; Martins *et al*., 2016), smart cities (Krimmling and Peter, 2014) to healthcare industry (Nachabe *et al*., 2015) where real-time updates of the patient's status were provided via. CoAP.

Background: In this study we describe the two publish/subscribe protocols dedicated to resource constrained networks in particular, focused on the wireless sensor networks. The first protocol is Message Queuing Telemetry Transport (MQTT) and the second is Constrained Application Protocol (CoAP).

Both CoAP and MQTT implement a lightweight application layer, as shown in Fig. 1 where an overview of the two protocols architecture and the support communication of data generated from sensors to the cloud and smartphones is drawn.

In fact in the literature, there are several works in which MQTT and CoAP protocols are together evaluated like by Mijovic *et al*. (2016) where researchers present a comparative study of these two protocols via. a real experimentation. By Thangavel *et al*. (2014) researchers present an evaluation of MQTT and CoAP performance via. a common middleware. However, a study of industrial protocols in the IoT including MQTT and CoAP is presented in Foster. Furthermore by Billavista and Zanni (2016) researchers propose a combined exploitation of MQTT and CoAP in order to achieve better scalability.

By Collina *et al*. (2014), MQTT and CoAP protocols are analyzed comparatively in terms of latency and throughput. Researchers have exploited a platform called Ponte in order to simulate different network characteristics as packet loss, delay and bandwidth limitations.

In this study, we will describe each protocol separately then we will proceed to a critical comparative study of some characteristics characterizing MQTT and CoAP and we will evaluate their performances in terms of efficiency and latency according to different metrics.

MATERIALS AND METHODS

MQTT overview: MQTT was proposed by OASIS to support IoT communications. It is a lightweight messaging protocol oriented to be used in resource-constrained devices and Machine-to-Machine (M2M) interactions in the mobile sector. MQTT is an application layer; it includes three components a subscriber, a publisher and a broker. It uses a topic-based publish-subscribe architecture as shown in Fig. 2.

To establish a communication between the three MQTT components, three interactions are required, subscribers should 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 this topics will receive the messages published via. the broker. This synchronisation is drawn in Fig. 3.

On the other hand, MQTT is like HyperText Transfer Protocol (HTTP), it relies on Transmission Control

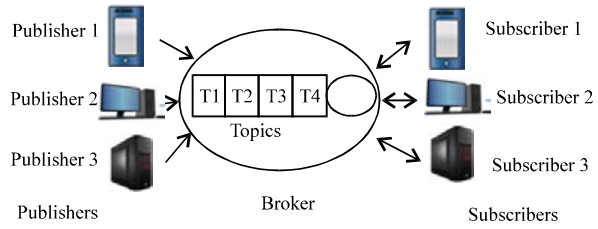

Fig. 2: MQTT protocol architecture

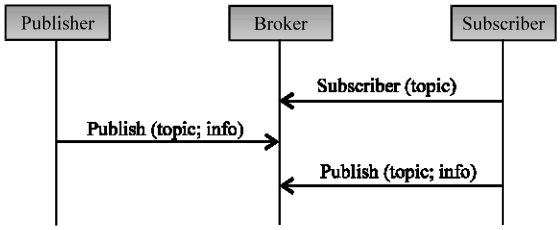

Fig. 3: The synchronization between the three components of MQTT

Protocol (TCP). However, compared to HTTP, MQTT is designed to have a lower protocol overhead. Additionally, MQTT is secure in terms of messages loss, i.e., even if the connection breaks off briefly, all messages are transmitted to clients, solving problems that arise upon unreliable communications (Luzuriaga *et al*., 2015a, b).

Nowadays, two main specifications exist for MQTT, MQTT V.3.1 (Locke, 2010) and MQTT-SN (also known as MQTT-S) (Stanford-Clark and Truong, 2013).

MQTT protocol provides a support for delivering messages between publishers and subscribers. QoS is an attribute of an individual MQTT message being published. So, to guarantee that a message has been received an acknowledgments exchange mechanism between the client and the broker is taken place. This mechanism is associated with a quality of service level specified on each message (Patierno, 2014).

QoS level zero (QoS = O): The sender sends the message only once and no retries are performed in other words, "Fire and forget". In this level messages are not necessarily delivered to their destination, this totally depends on the reliability of TCP/IP. If a TCP/IP session is broken, the messages are lost.

QoS level one (QoS = 1): MQTT ensures that a message arrives at its destination at least once. So, 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. If a TCP/IP session is broken and regarding the limited space of the buffer, it can store only a few messages until the time when the session is again restored and acknowledgment messages again are delivered.

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

Nevertheless, MQTT 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, researchers in (Luzuriaga *et al*., 2015b) propose an intermediate buffering and evaluate it in various scenarios where the publisher node suffers a handover process due to mobility. In the same context, a seamless handover for a hotspot network using a buffering technique is proposed by Yamagata *et al*. (2006) where researchers 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. However, by Luzuriaga *et al*. (2016) researchers propose a cross-layer solution and an 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.

In the following, we will present a CoAP overview then we will open a discussion around the CoAP limitations.

CoAP overview: CoAP has been designed by the Internet Engineering Task Force (IETF) to support IoT with lightweight messaging for constrained devices operating in a constrained environment. Like MQTT, CoAP is an application layer protocol but based on a REST architecture. In this architecture an application process affords resources to applications and a Universal Resource Identifiers (URI) defines these resources CoAP defines two kinds of interactions between end-points, the client/server and the publish/subscribe interactions. The client/server model supports two interaction types, a one-to-one interaction, i.e., request/reply and a multi-cast interaction, i.e., a client interrogating several servers using requests and the publish/subscribe model called the observer model, here, the server plays the role of a publisher and the observer plays the role of a

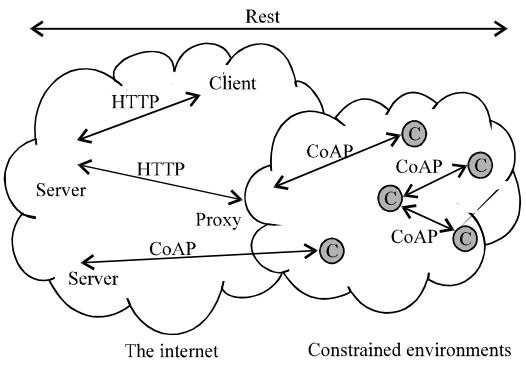

Fig. 4: An overview architecture of CoAP protocol

subscriber. A server can send messages of notifications called publications about an event interesting the observer.

Unlike MQTT and HTTP, CoAP doesn't run over TCP, it runs over UDP. Communication between clients and servers is afforded through connectionless datagrams. Retries and reordering are implemented in the application stack. UDP broadcast and multicast are also allowed by CoAP for addressing (Jaffey, 2014). Otherwise, 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 message arrived without the significant communication overhead as in the case of TCP (Masek *et al*., 2016). Overview architecture of CoAP protocol is drawn in Fig. 4.

In addition, CoAP utilizes four message types; confirmable, non-confirmable, reset and acknowledgment where two of them concern reliability messages. The reliability of CoAP is simple compared to MQTT quality of service, it consists of a confirmable message and a non-confirmable message (Davis *et al*., 2013). In the case of confirmable message an Acknowledgment message (ACK) is sent to the sender from the intended recipient, else the message is retransmitted. This is just a confirmation that the message is received but it doesn't confirm that its contents were decoded correctly. However, a non-confirmable message is fire and forget, i.e., no reception confirmation.

CoAP has become increasingly proposed for gathering data from smart sensors and controlling constrained devices. However, CoAP needs more requirements to compete with older, traditional network management protocols which allow realizing very similar functions such as SNMP (Stallings, 1999) and NETCONF.

As MQTT, the mobility of device represents the great limitation that hinders the proper functioning of CoAP protocol and causes the loss of packets. In order to overcome this limitation, researchers in (Choi and Koh, 2016) propose to use a proxy mobile IPv6 for mobility management. However, in Chun and Park (2015) a mobile CoAP for mobility management (CoMP) is presented. Here, researchers propose to keep track of the current mobile sensor IP addresses during the handover and using both HTTP and CoAP sensed data is reliably delivered to the Web clients.

Furthermore, the second limitation to quote in CoAP running is the network congestion. Even if the core CoAP specification defines a basic congestion control mechanism to make it able to handle congestion control by itself, CoAP is not capable of adapting to network conditions. That the reason why researchers in Betzler *et al*. (2016) exhibit CoCoA; an advanced congestion control mechanism for CoAP.

Summary: In the process of subscription-publication, the challenge is to accomplish the message delivery with a high efficiency, a low latency and a low packet loss rate using one of reliability and QoS level. Otherwise, it is up to the application to select the appropriate QoS level for its publications and subscriptions, thus the decision to use one of these levels impacts on the application performance as well as on the use of bandwidth and battery life of devices.

However, although the traditional protocol's effectiveness, the need for a suitable protocol for IoT applications involving constrained devices is necessary because the biggest obstacle in the functioning of these devices is energy consumption (Johnson, 2016). Furthermore, regarding the fact that devices are expected to continue functioning for many years, researchers are urged to come up with appropriate solutions. In the following, a comparative study of MQTT and CoAP is drawn.

Comparison: MQTT and CoAP are rapidly emerging and integrating the IoT market as leading lightweight messaging protocols for constrained devices. Each protocol offers unique benefits and each poses challenges and tradeoffs. The strengths and issues of these two protocols are summarized in Table 1. A summary of the key criteria considered in this study is drawn in Table 2.

Table 1: Strengths and issues in both MQTT and CoAP (Stansberry, 2016)

Strengths	Issues
MQTT	
Publish/subscribe model	Central broker
MQTT's pub/sub model offers scalability and good performances	TCP
Space decoupling	The connection mode of TCP requires a persistent session between communicating nodes, however regarding the energy of resource-constrained devices, this persistence becomes difficult.
Nodes get information directly from the central broker without having any knowledge of the sender node	Wake up time
Time decoupling	The process of connection of TCP which precede exchanging messages is multi-step. This increases communication times and reduces battery life, especially for nodes with periodic or repetitive traffic
Nodes don't have to be active while others are publishing.	
Synchronization decoupling	
Data received from a node in full operation is queued by the broker until the receiving node has finished	
Flexible topic subscriptions	
An MQTT node can subscribe to all messages concerning a given event.	
A broker dysfunction led to a network inoperability	
CoAP	
UDP	Standard maturity (complexity)
Connectionless datagrams enable devices to conserve energy as it will sleep for a long period of time	It is easier to run an MQTT network than a similar network using CoAP. Indeed CoAP presents several interoperability challenges.
Multi-cast support	Message reliability (QoS level)
A CoAP network supports one-to-one, one-to-many or many-to-many function.	It is simple but not trustworthy, since, confirmable message confirms that the message is received but without confirmation that the message was correctly decoded. However, non-confirmable messages is fire and forget
Asynchronous communication	
CoAP has a simplified observer mechanism similar to MQTT broker but it doesn't need the same requirements. The observer enables nodes to observe others without actively engaging them	

Table 2: Summary of key comparison criteria

Variables	MQTT	CoAP
Abstraction	Pub/Sub	Request/Reply
Architecture style	Brokered	P2P
QoS	At most once	Confirmable messages
	At least once	non-confirmable messages
	Exactly one	
Interoperability	Partial	Yes
Transports	TCP	UDP
Data serialization	Undefined	Configurable
Standards	Proposed OASIS MQTT standard M	Proposed IETF CoAP standard
Mobile devices (Android, iOS)	Yes	Via. HTTP proxy
6LoWPAN devices	Yes	Yes
Dynamic discovery	No	Yes
Security	Username/Password Authentication	
	SSL for data encryption	DTLS for data encryption

RESULTS AND DISCUSSION

Performance evaluation: Both 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 each protocol depends on the performances resulted in the scenarios and the experiment conditions used.

In order to evaluate the performances of MQTT and CoAP in terms of throughput, latency and inter-arrival time we perform in this study, emulations. Using CORE network emulator, we draw two scenarios, low data traffic and high data traffic. Furthermore, two metrics were considered under each scenario, low link packet loss and high link packet loss. The parameters considered in this emulation are detailed in Table 3.

Table 3: Emulation parameters

Parameters	Values
Data traffic	1 node (100 messages), 10 nodes (10^4 messages)
Message size	6 and 8 kB
Link packet loss (PL)	0.01, 0.1
Time	0 and 100 sec

Low data traffic scenario: Figure 5 and 6 show the results of low traffic data scenario using a single node. Under the two conditions of low link packet loss (Fig. 5) and high link packet loss (Fig. 6), results show that MQTT performs well in terms of throughput and present a lower inter-arrival time compared to CoAP, however CoAP performs better in terms of latency, thing that has increased dramatically from low to high link packet loss in MQTT. These results are explained by the fact that MQTT and CoAP use different transmission protocols of TCP and UDP.

High data traffic scenario: In this scenario, we use multiple nodes in order to afford a higher traffic. According to results shown in Fig. 7 both MQTT and CoAP throughput have increased proportionally compared to low traffic scenario but MQTT still performs well. The degradation of CoAP throughput is due to its retransmission mechanism. In terms of latency and inter-arrival, MQTT and CoAP present slightly the same performances. Otherwise, in Fig. 8, results indicate that both MQTT and CoAP throughput has decreased slightly compared to the same scenario using low link packet loss but CoAP present the best performances in terms of both throughput and latency. Finally, in terms of inter-arrival MQTT and CoAP present slightly the same performances.

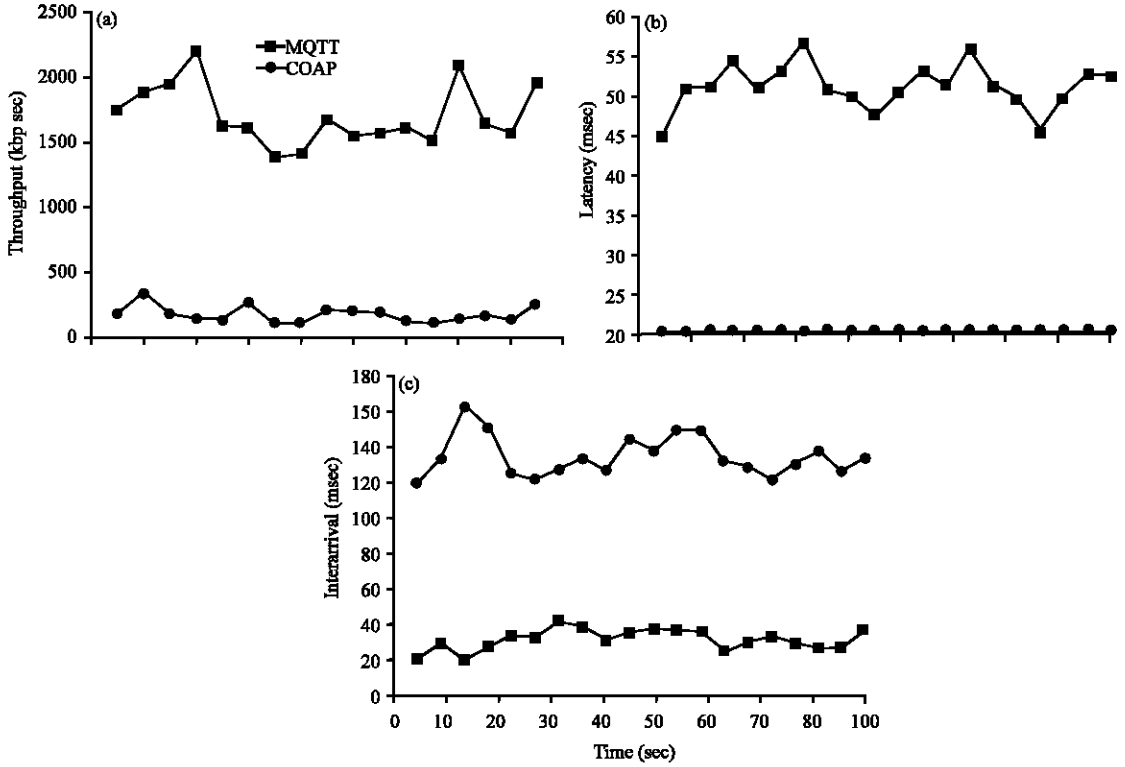

Fig. 5: Throughput: a) Latency; b) Inter-arrival and c) On low data traffic scenario under low link packet loss condition

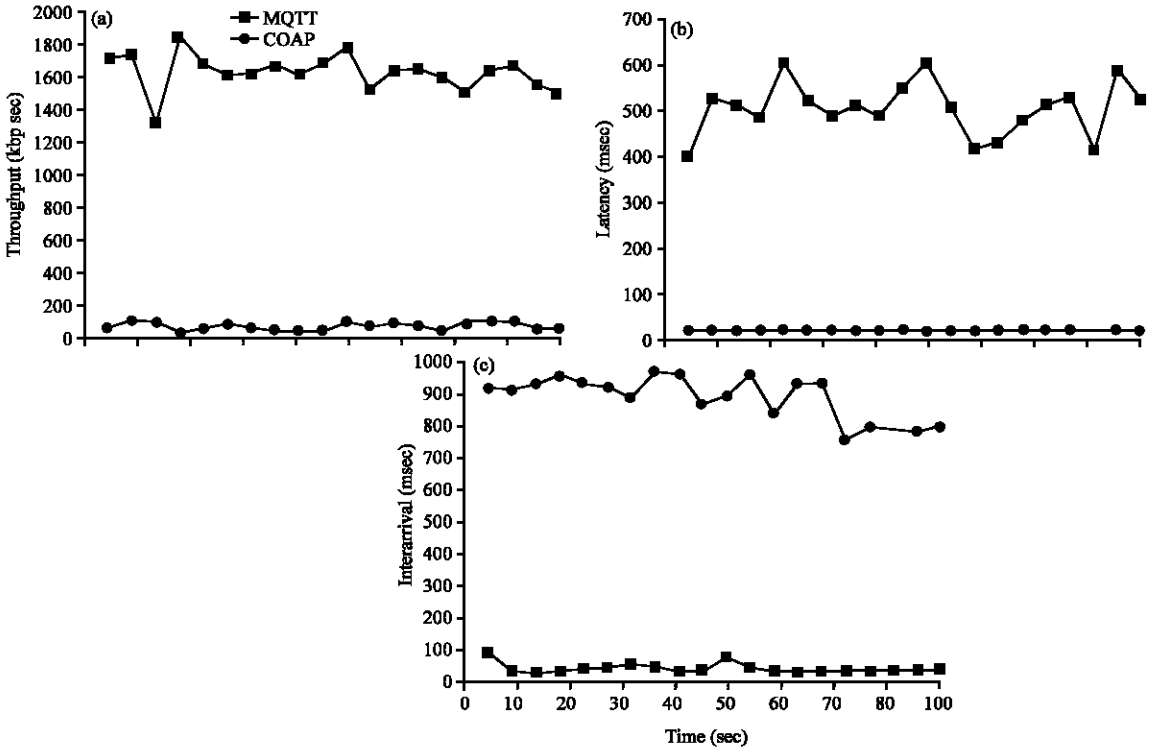

Fig. 6: Throughput: a) Latency; b) Inter-arrival and c) On high data traffic scenario under high link packet loss condition

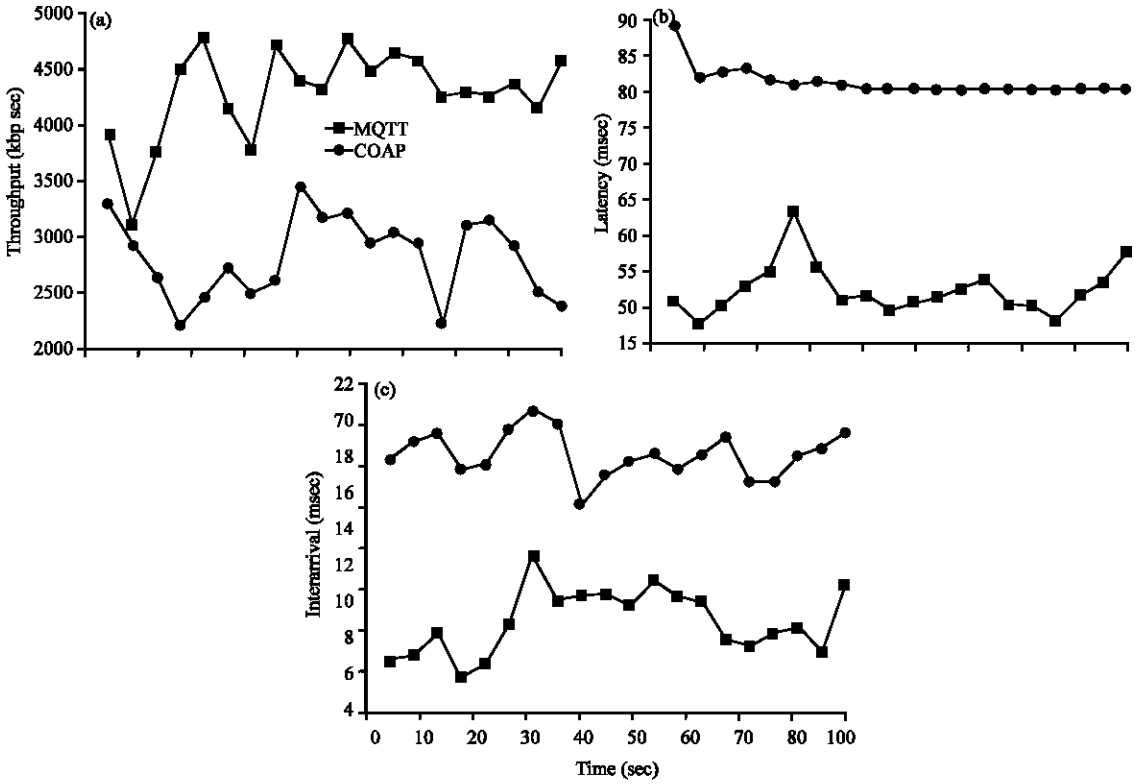

Fig. 7: Throughput: a) Latency; b) Inter-arrival and c) On high data traffic scenario under low link packet loss condition

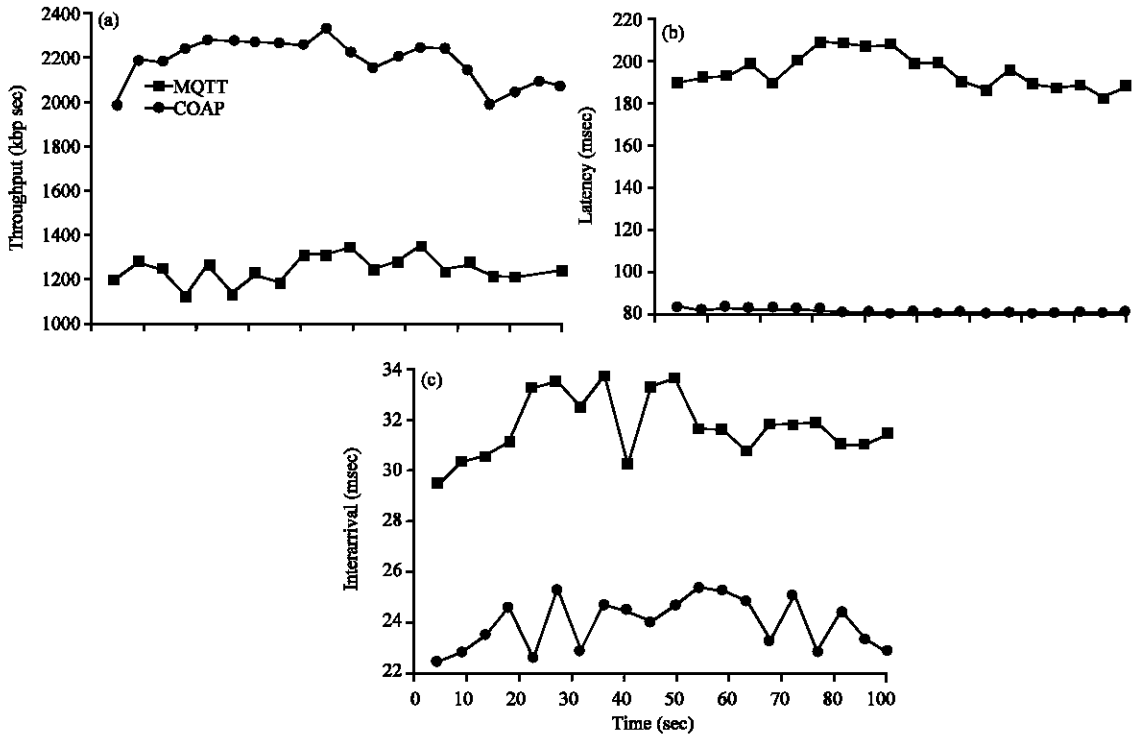

Fig. 8: Throughput: a) Latency; b) Inter-arrival and c) On high traffic scenario under high link packet loss condition

The purpose of this evaluation is to choose the use of either MQTT or CoAP according to the best throughput and lowest latency resulted in the presence of different metrics (delay, link packet loss and data traffic offered by the devices).

So, in the choice of the most suitable messaging solution for an application, the study of the choice should not only be based on an understanding of the architecture but also the main system requirements in terms of interoperability and performances as emulations results have proved.

CONCLUSION

Thanks to the Internet of Things (IoT), it is now possible to interconnect objects to transfer data and to publish information over a network without requiring human-to-human or human-to-computer interaction. The IoT is connecting different devices in our entourage using WSN and based on different available connection models. The challenge is each of which can be used to connect constrained devices in a distributed network. That is why the IoT application programmers are faced with the challenges of choosing an appropriate communication protocol for their resource-constrained applications. So, in order to assist programmers in their decision process, this study presents a comparative study of two of the most important protocols based on Pub/Sub Model and designed for constrained devices, MQTT and CoAP. The comparison is drawn in terms of security, quality of service and performances. Finally, as with all new protocol deployments, work is underway to continue improving MQTT and CoAP protocols in order to achieve scalability and best performances in terms of efficiency and latency of communications.

REFERENCES

Anonymous, 2014. MQTT v3.1.1 now an OASIS standard. MQTT, USA. http://mqtt.org/

Bellavista, P. and A. Zanni, 2016. Towards better scalability for IoT-cloud interactions via combined exploitation of MQTT and CoAP. Proceedings of the 2016 IEEE 2nd International Forum on Research and Technologies for Society and Industry Leveraging a better tomorrow (RTSI'16), September 7-9, 2016, IEEE, Bologna, Italy, ISBN:978-1-5090-1132-2, pp: 1-6.

Bergmann, O., K.T. Hillmann and S. Gerdes, 2012. A CoAP-gateway for smart homes. Proceedings of the 2012 International Conference on Computing, Networking and Communications (ICNC'12), January 30-February 2, 2012, IEEE, Maui, Hawaii, ISBN:978-1-4673-0008-7, pp: 446-450.

Betzler, A., C. Gomez, I. Demirkol and J. Paradells, 2016. CoAP congestion control for the internet of things. IEEE. Commun. Mag., 54: 154-160.

Choi, S.I. and S.J. Koh, 2016. Use of proxy mobile IPv6 for mobility management in CoAP-based internet-of-things networks. IEEE. Commun. Lett., 20: 2284-2287.

Chun, S.M. and J.T. Park, 2015. Mobile CoAP for IoT mobility management. Proceedings of the 12th Annual IEEE Conference on Consumer Communications and Networking (CCNC'15), January 9-12, 2015, IEEE, Las Vegas, Nevada, ISBN:978-1-4799-6389-8, pp: 283-289.

Collina, M., M. Bartolucci, A. Vanelli-Coralli and G.E. Corazza, 2014. Internet of Things application layer protocol analysis over error and delay prone links. Proceedings of the 7th International Conference on Advanced Satellite Multimedia Systems and the 13th Workshop on Signal Processing for Space Communications (ASMS/SPSC), September 8-10, 2014, IEEE, Livorno, Italy, ISBN:978-1-4799-5893-1, pp: 398-404.

Davis, E.G., A. Calveras and I. Demirkol, 2013. Improving packet delivery performance of publish/subscribe protocols in wireless sensor networks. Sens., 13: 648-680.

Eugster, P.Th., P.A. Felber, R. Guerraoui and A.M. Kermarrec, 2003. The many faces of publish/subscribe. ACM Comput. Surv., 35: 114-131.

Jaffey, T., 2014. MQTT and CoAP, IoT protocols. Eclipse Foundation, Ottawa, Canada. https://www.eclipse.org/community/eclipse_newsletter/2014/february/article2.php

Johnson, S., 2016. Constrained application protocol: CoAP is IoT's 'modern' protocol. https://internetofthingsagenda.techtarget.com/feature/Constrained-Application-Protocol-CoAP-is-IoTs-modern-protocol

Krimmling, J. and S. Peter, 2014. Integration and evaluation of intrusion detection for CoAP in smart city applications. Proceedings of the 2014 IEEE Conference on Communications and Network Security (CNS'14), October 29-31, 2014, IEEE, San Francisco, California, ISBN:978-1-4799-5890-0, pp: 73-78.

Locke, D., 2010. MQ Telemetry Transport (MQTT) V3.1 protocol specification. IBM, Armonk, New York, USA. https://www.ibm.com/ developerworks/library/ws-mqtt/index.html

Luzuriaga, J.E., J.C. Cano, C. Calafate, P. Manzoni and M. Perez *et al*., 2015a. Handling mobility in IoT applications using the MQTT protocol. Proceedings of the Internet Technologies and Applications (ITA'15), September 8-11, 2015, IEEE, Wrexham, Wales, UK., ISBN:978-1-4799-8036-9, pp: 245-250.

Luzuriaga, J.E., M. Perez, P. Boronat, J.C. Cano and C. Calafate *et al*., 2015b. A comparative evaluation of AMQP and MQTT protocols over unstable and mobile networks. Proceedings of the 12th Annual IEEE Conference on Consumer Communications and Networking (CCNC'15), January 9-12, 2015, IEEE, Las Vegas, Nevada, ISBN:978-1-4799-6389-8, pp: 931-936.

Luzuriaga, J.E., M. Perez, P. Boronat, J.C. Cano and C. Calafate *et al*., 2016. Improving mqtt data delivery in mobile scenarios: Results from a realistic testbed. Mob. Inf. Syst., 2016: 1-11.

Martins, R.D.J., V.G. Schaurich, L.A.D. Knob, J.A. Wickboldt and A.S. Filho *et al*., 2016. Performance analysis of 6LoWPAN and CoAP for secure communications in smart homes. Proceedings of the 2016 IEEE 30th International Conference on Advanced Information Networking and Applications (AINA'16), March 23-25, 2016, IEEE, Mollens, Valais, Switzerland, ISBN:978-1-5090-1857-4, pp: 1027-1034.

Masek, P., J. Hosek, K. Zeman, M. Stusek and D. Kovac *et al*., 2016. Implementation of true IoT vision: Survey on enabling protocols and hands-on experience. Intl. J. Distrib. Sens. Netw., 12: 1-18.

Maynard, N., 2011. Using websphere MQ telemetry and pachube to connect to remote sensors and devices. IBM DeveloperWorks, USA. https://www.ibm.com/developerworks/websphere/library/techarticles/1106_maynard/1106_maynard.html

Mijovic, S., E. Shehu and C. Buratti, 2016. Comparing application layer protocols for the Internet of Things via experimentation. Proceedings of the 2016 IEEE 2nd International Forum on Research and Technologies for Society and Industry Leveraging a Better Tomorrow (RTSI), September 7-9, 2016, IEEE, Bologna, Italy, ISBN:978-1-5090-1132-2, pp: 1-5.

Nachabe, L., M. Girod-Genet, B. El-Hassan and J. Jammas, 2015. M-health application for neonatal incubator signals monitoring through a CoAP-based multi-agent system. Proceedings of the 2015 International Conference on Advances in Biomedical Engineering (ICABME'15), September 16-18, 2015, IEEE, Beirut, Lebanon, ISBN:978-1-4673-6515-4, pp: 170-173.

Patierno, P., 2014. MQTT & IoT protocols comparison. LinkedIn Corporation, Sunnyvale, California, USA., https://www.slideshare.net/paolopat/mqtt-iot-protocols-comparison

Prada, M.A., P. Reguera, S. Alonso, A. Moran and J.J. Fuertes *et al*., 2016. Communication with resource-constrained devices through MQTT for control education. IFAC. Pap. Online, 49: 150-155.

Robinson, J., G.F. Jeremy, J.S. Andy, A.D. Reynolds and B.V. Bedi, 2005. Sensor networks and grid middleware for laboratory monitoring. Proceedings of the 1st International Conference on e-Science and Grid Computing, Dec. 5-8, Melbourne, Australia. IEEE Computer Society, pp: 569-577.

Shin, I.J., D.S. Eom and B.K. Song, 2015. The CoAP-based M2M gateway for distribution automation system using DNP3. 0 in smart grid environment. Proceedings of the 2015 IEEE International Conference on Smart Grid Communications (SmartGridComm'15), November 2-5, 2015, IEEE, Miami, Florida, ISBN:978-1-4673-8288-5, pp: 713-718.

Stallings, W., 1999. SNMP, SNMPv2, SNMPv3 and RMON 1 and 2. 3rd Edn., Addison-Wesley, Boston, Massachusetts, USA., ISBN:9780201485349, Pages: 619.

Stanford-Clark, A. and H.L. Truong, 2013. MQTT for Sensor Networks (MQTT-SN): Protocol specification Version 1.2. IBM, Armonk, New York, USA. http://mqtt.org/new/wp-content/uploads/2009/06/MQTT-SN_spec_v1.2.pdf

Stansberry, J., 2015. MQTT and CoAP: Underlying protocols for the IoT. Electronic Design, USA. https://www.electronicdesign.com/iot/mqtt-and-coap-underlying-protocols-iot

Thangavel, D., X. Ma, A. Valera, H.X. Tan and C.K.Y. Tan, 2014. Performance evaluation of MQTT and CoAP via a common middleware. Proceedings of the IEEE 9th International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), April 21-24, 2014, IEEE, Singapore, ISBN:978-1-4799-2843-9, pp: 1-6.

Upadhyay, Y., A. Borole and D. Dileepan, 2016. MQTT based secured home automation system. Proceedings of the 2016 Symposium on Colossal Data Analysis and Networking (CDAN'16), March 18-19, 2016, IEEE, Indore, India, ISBN:978-1-5090-0670-0, pp: 1-4.

Yamagata, F., D. Komaba, H. Oguma, S. Kameda and H. Nakase *et al*., 2006. Seamless handover for hotspot network using buffered packet forwarding method. Proceedings of the 10th IEEE International Conference on Communication Systems (ICCS'06), October 30-November 1, 2006, IEEE, Singapore, pp: 1-5.

Zhang, L., 2011. Building Facebook messenger. Facebook, Inc.,California, USA. https: //web. facebook. com/notes/facebook-engineering/building-facebook-messenger/ 10150259350998920/?-rdc = 1&-rdr