

A Review of Application Protocol Enhancements for Internet of Things

Fatma Hmissi
 Cristal Laboratory, ENSI
 Manouba University Campus
 Manouba 2010, Tunisia
 e-mail: fatma.hmissi@ensi-uma.tn

Sofiane Ouni
 Cristal Laboratory, ENSI
 Manouba University Campus
 Manouba 2010, Tunisia
 e-mail: soufiane.ouni@insat.rnu.tn

Abstract—Nowadays, all sectors utilize devices that are part of the Internet of Things (IoT) for the purpose of connecting and exchanging information with other devices and systems over the Internet. This increases the diversity of devices and their working environments, which, in turn, creates new challenges, such as real-time interaction, security, interoperability, performance, and robustness of IoT systems. To address these, many applications protocols were adopted and developed for devices with constrained resources. This paper surveys communication protocols divided according to their goals along with their merits, demerits, and suitability towards IoT applications. We summarize the challenges of communication protocols as well as some relevant solutions.

Index Terms—Internet of Things (IoT), Messaging Protocol, Device Management Protocol, Service Discovery Protocol, Constrained devices, Interoperability, Security, Quality of Service (QoS).

I. INTRODUCTION

The Internet of Things (IoT) refers to the physical objects that are connected to the Internet. The IoT is used to make people's lives and businesses easier in many areas. Generally speaking, there is no standard architecture for the Internet of Things systems, but what is certain is that all architectures are composed of several parts which interact and communicate with each other without human intervention. The Internet of Things connects the real world of things to the virtual world of networks and the cloud.

Figure 1 introduces a typical scenario of the IoT system where the interaction between the different parts is clearly presented. The Devices are the most important elements in the system. They are equipped with embedded sensors, actuators, and controllers to interact with the physical environments to collect information or to change the actual status. A device can exchange data either with other devices or with data-center, the Cloud, or other servers. The Gateway represents a physical entity that is composed of several electronic devices. The main purpose of the Gateway is to connect to a different networks having different typologies. It contains software that translates the protocols to establish communication between the things and the network. The number of connected devices is expected to grow rapidly, with a predicted 75 billion devices worldwide expected to be connected to the Internet by 2025 [1]. This great number of connected

devices is expected to generate unlimited data. As a result, an enormous amount of data to be stored, processed, and made available in a continuous, efficient, and easily interpretable manner is growing rapidly, which puts a lot of pressure on the Internet infrastructure. To solve this problem, companies combined the capabilities of IoT and cloud computing. The technology of cloud computing assists in alleviating the pressure on the Internet infrastructure by storing, processing, and transferring data to the Cloud instead of to the connected devices. Many platforms, called IoT Cloud Platforms, exploit Cloud Computing features to provide IoT services. For this purpose, a number of open sources and proprietary IoT platforms have been proposed and implemented to provide many efficient and easy IoT services, such as data collection, storage, analysis, monitoring, control, and management of connected things. Today, more than 300 IoT platforms are available on the market [2]. Mobile and Web applications make the IoT very user-friendly. A mobile application is a software application that is created to run on mobile devices especially those that are small and wireless. A Web application is a software application that is hosted on a server and accessible through a Web browser. Mobile and Web applications allow users to perform a set of specific functions and tasks on the Internet. These functions and tasks are summarized in the connection, monitoring, control, and management of connected objects.

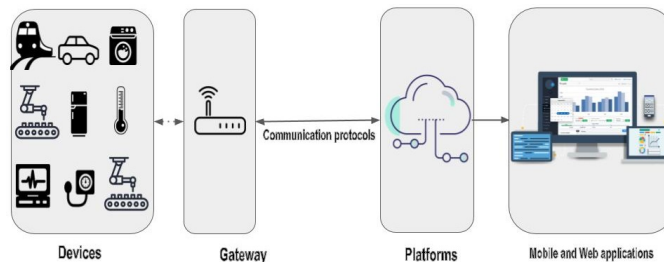


Fig. 1. IoT Basic Architecture.

Communication Protocols are used to allow the connectivity for data exchange between physical or virtual entities, e.g., Devices and the Cloud, by defining rules and constraints where several requirements must be taken into account by these pro-

protocols in order to be successful within the IoT systems. Those protocols are divided into three categories. This classification is based on the purposes of the protocols. It is possible to build an IoT system with the typical communication protocols based on classical HTTP Web requests even if they do not have certain requirements. However, they are no longer the right choice for Internet of Things, for this purpose, other messaging protocols (such as Message Queuing Telemetry Transport (MQTT) [3] [4], Constrained Application Protocol (CoAP) [5] [6], Data Distribution Service (DDS) [7] [8] [9] [10]) have been implemented. It is seen that these messaging protocols are the best way to create an interconnection between heterogeneous devices and IoT systems, however, this is not sufficient to resolve all connectivity issues. Indeed, a huge number of connected devices are deployed in remote, hostile and hard to reach places, which makes their configurations and maintenance are difficult. Many solutions are proposed to provide device management necessity. For example, Perumal et al. [11] proposed a lightweight IoT device management framework for smart home services. Mechanisms for discovery are important to use the services of the Internet of Things. Service discovery is a process of automatically locating the appropriate services. Ahmed et al. [12] proposed a secured service discovery technique for the Internet of Things.

To sum up, application protocols are subdivided into three types, namely, (i) for messaging, (ii) for device management, and (iii) for service discovery.

The weaknesses of the current solutions have motivated the improvement of the existing protocols that seek to enhance the Internet of Things ecosystems' performance and avoid faults.

Thus, this work aims at presenting an extensive survey about the existent communications protocols that can be used in IoT applications. Different from the current existing surveys in the literature, this work does not only consider existing and well-known base protocols, but also all relevant solutions that have been introduced during recent years. The main contributions of this paper are summarized as follows:

- Overviews of the initial communication protocols adapted for application.
- Overviews of the initial communication protocols adopted for device management.
- Overviews of the initial communication protocols adopted for service discovery.
- Identifies the problems most studied by the existing protocols in IoT scenarios.
- Reviews of the studied solutions that improve existing protocols.

The remainder of this paper is organized as follows: Section II presents the communication protocols for the Internet of Things applications. Section III lists the communication protocols for device management. Section IV sums up the communication protocols for service discovery. Section V introduces the challenges for the communication protocols and surveys the recent approaches to the protocols enhancement. Section VI concludes the paper.

II. MESSAGING PROTOCOLS

IoT cannot rely on a single protocol for all needs [13]. Consequently, several of available messaging protocols are chosen for various types of requirements of the IoT system [14]. Thus, in the rest of this section, the most relevant protocols are cited with their descriptions.

Message Queuing Telemetry Transport (MQTT) is a lightweight [3] [4] and flexible [15] messaging protocol. MQTT uses different approaches for routing mechanisms, such as one-to-one, one-to-many, or many-to-many, making the connection between IoT and Machine-to-Machine (M2M) to connected devices/applications possible [15]. M2M is used to provide communications between machines without human intervention. MQTT is designed as a publish-subscribe model [15], using TCP as transport layer protocol. The process of message exchange in MQTT consists of multiple clients connected to a central broker which is a server running somewhere in the Internet network [16]. These clients' act as senders that publish a message on a particular topic that could be considered as message subject [15] or as a receiver that is subscribed to a topic registered at the broker. Every time the MQTT Broker gets a new publish message to a specific topic, it broadcasts this message to the entire subscribed receivers. The MQTT protocol is used between devices and Cloud, but it cannot be used between devices. Another disadvantage is that MQTT does support multi-cast.

Constrained Application Protocol (CoAP) [5] [6] is mainly used in a constrained environment with constrained devices and constrained networks. It uses unicast and multi-cast request-response model for interaction between multiple clients and multiple servers by sending request and response messages using a URI with GET, POST, PUT and DELETE actions over UDP to keep things lightweight. The main merit of the CoAP protocol is that it can be used with constrained devices in interaction device-to-device. This protocol cannot be used in asynchronous communication because it does not support publisher-subscriber architecture. Also, it does not support broadcast. The clients cannot use a topic to send and respond to messages.

Data Distribution Service (DDS) [7] [8] [9] [10] is used for real-time and industrial M2M communications, running over both TCP or UDP. DDS supports broker-less architecture where it uses a publish-subscribe model for interaction between entities without the use of a Broker. The tasks of a broker are handled by Data Writers (DW) and Data Readers (DR). The main advantages of DDS protocol are that the data usage is fundamentally anonymous, since the publishers do not enquire about who consumes their data, and the probability of system failure is limited (system more reliable) because there is no single point (no broker) of failure for the entire system [7]. The most remarkable disadvantage of DDS is that it is designed for Industrial application (IIoT) with considerable hardware resources. This makes the implementation for constrained devices that need a Lightweight protocol even

harder. The other disadvantage is related to the increase of the communication workload by the publishing of data even if there are no interested subscribers [7].

Advanced Message Queuing Protocol (AMQP) [7] [8] is designed as a publish-subscribe model which uses TCP as transport layer protocol. Mainly, it has three components, Publishers, Subscribers and, both parts of an AMQP Broker are Exchanges of Message queues. The Publisher creates a bare message and sends it to the Exchanges components that are used to forward the messages to appropriate message queues using the routing keys contained in messages. The latter can be stored into message queues before forwarding them to Subscribers. If there are more subscribers interested in a particular message, the broker can duplicate the messages and send their copies to multiple queues waiting for annotated messages from subscribers. The main advantage of the AMQP protocol is that it could be used in device-to-device, device-to-Cloud, and Cloud-to-Cloud interaction. But its main disadvantage is that the publishers and subscribers cannot publish and subscribe using the topic.

As introduced in [17] [18] [19], the eXtensible Messaging and Presence Protocol (XMPP), also known as Jabber, is a standard initially designed for instant messaging and exchange of messages between applications no matter which operating system they are using in IoT. It is designed to allow users to send messages in real-time and manage the presence of the user. XMPP supports Publish-Subscribe and Request-Response models with TCP transport protocol. To exchange messages between clients and servers XMPP uses streams of stanzas. XMPP is a text-based protocol where XMPP stanzas [17] [18] [19] are Extensible Markup Language (XML) messages exchanged between clients. The main advantage of the AMQP protocol is that could be used in device-to-device, device-to-Cloud interaction. As AMQP, DDS, and CoAP, topics are not used to publish and subscribe with XMPP.

IoT messaging protocols are designed for communication. All the cited protocols lack IoT device management and service discovery procedure. Our depth study allows us to conclude that the messaging protocol with the greatest impact on the IoT is the MQTT. The MQTT protocol is the lightest, the most robust and the one that has the least overhead.

III. DEVICE MANAGEMENT PROTOCOLS

A huge amount of heterogeneous devices, which are integrated into IoT, need to be (re)discovered, reconfigured, and maintained to fix security issues, deploy new features, or recover from their failures. It is possible to manage devices with the IoT messaging protocols by inventing new building blocks. It seems that these protocols are no longer the right choice for device management because of the high cost of development, where for every new management feature, a new block should be developed. To solve this problem, new protocols known as Device Management Protocols are proposed. A device management protocol enables the abstraction

of an IoT/M2M device as a managed object to make the management of the device much easier [20].

Open Mobile Alliance Device Management standard [21] [22] [23] named as OMA-DM and designed by Open Mobile Alliance for device management, is used for Terminal M2M devices and Mobile terminal devices, e.g., Mobile phones, Smartphones, Tablets, laptops. Mobile network operators and enterprises use OMA-DM to manage mobile devices remotely. The main features of OMA-DM are: read and write configuration or monitoring nodes, read and set parameter keys and values, Firmware Update Management Object (FUMO), software components management object (SCMO) that means install, upgrade, or uninstall software elements. OMA-DM has several demerits. The OMA-DM protocol is designed only for no constrained and fixed devices. Another disadvantage is that OMA-DM cannot be used for industrial applications and cannot be built on top of the MQTT protocol. The disadvantage before the last is that it supports the only XML serialization format and it does not support either Binary, Plain text, or TLV and JSON serialization format. The final disadvantage refers to the no support of interoperability.

Lightweight M2M [24] [25] (LwM2M) is a client-server standard developed by the Open Mobile Alliance (OMA). It is an OMA-DM successor. The LwM2M is a standard device and service management built on top of CoAP to ensure remote management and configuration of constrained and powerful devices. It can benefit from efficient communication in M2M and IoT environments over UDP and SMS bearers. So, SMS can be used for waking up the device or any GET, POST, and PUT request. The LwM2M main features are: device monitoring and configuration, server provisioning (bootstrapping) and firmware upgrades. There are numerous advantages of the LwM2M protocol. The most remarkable advantage of LwM2M is that it could be used with fixed and mobile-constrained devices. Another advantage refers to the support of the industrial application and interoperability. The most important disadvantages of LwM2M are: cannot support XML serialization format, cannot be built on top of MQTT and cannot be used in telecommunication applications.

The Broadband Forum defined CPE WAN management protocol (CWMP) that is used for remote management of home and business network devices, such as modems, gateways, routers, and VOIP phones (see Technical report 069 [26] [27] known as TR-069). The main capabilities of this protocol are firmware management, auto-configuration, dynamic service provisioning, software module management, status monitoring, performance monitoring, and diagnostics. The TR-069 uses SOAP/HTTP protocol for communication between network devices called the Customer Premises Equipment (CPE) and central server called the Auto-Configuration Servers (ACS). The CPE and ACS present the main components of this protocol. TR-069 has the same disadvantages as OMA-DM.

IoT devices management protocols are not oriented for communication and service discovery features. Our depth study

allows us to conclude that the device management protocol with the greatest impact on the IoT is the LwM2M. The overhead, footprint, and server load of the LwM2M are lighter than TR-065 and OMA-DM protocols, while the response time of LwM2M is faster than TR-065 and OMA-DM protocols.

IV. SERVICES DISCOVERY PROTOCOLS

Service Discovery Protocols (SDPs) are communication protocols that provide mechanisms to help clients to discover services available on the network. There are several SDPs for the IoT environment. This section focuses on the most known SDPs by introducing the following protocols: mDNS, SSDP. Multicast Domain Name System (mDNS) [28] [29] [30] is an open protocol defined by IETF which requires minimal configuration, based on the Internet Protocol (IP) and the User Datagram Protocol (UDP). An mDNS client can discover a thing's endpoint by resolving its hostname to an IP address. An mDNS client has to send an IP multi-cast query message over the network. The message calls the host with that name to reply and identify. Once the host receives the message, it replies via a multi-cast message that contains its IP address. All nodes in the network receiving that multi-cast message update their mDNS caches accordingly. This protocol, coupled with DNS-based Service Discovery (DNS-SD), offers the flexibility required by environments where it is necessary to automatically integrate new devices and perform DNS-like operations without the presence of a conventional DNS server.

The Simple Service Discovery Protocol (SSDP) [28] [29] [30] is an open protocol, based on IP, UDP, and SOAP (Simple Object Access Protocol) [28] [29] [30]. An SSDP client discovers SSDP services by multi-casting a discovery request to the SSDP multicast channel and port. SSDP services listen on that channel until they receive a discovery request that matches the service they offer, then they respond using a unicast response. This protocol—included in the Universal Plug-and-Play (UPnP) architecture—makes it possible to transparently plug and play devices without the need for any manual configuration.

V. CHALLENGES AND ENHANCEMENTS OF COMMUNICATION PROTOCOLS

Throughout the latest studies, it has been shown that protocols limitations and drawbacks do exist. Among these, we highlight communication protocols challenges:

- Real-time and industrial communication issues.
- Not suitable for constrained devices.
- Interoperability issues.
- Security issues.
- Quality of Service (QoS) issues.

Motivated by the presented issues, several new solutions have emerged recently. In this section, an overview of studies focusing on the improvement of existing and well-known base protocols are divided and presented according to their proposals. Table I summarizes the existing studies of

some widely efficient and recently enhanced approaches for application layer protocols in IoT environment.

Several applications in IoT fields, such as medical, factory, and transportation are time-sensitive applications. Mostly, the delays of communications between the different parts of the IoT systems are in-bounded. Therefore, the real-time requirement is one of the challenges of communication protocols. Most IoT solutions involve time constraints to gather and process information, make decisions, and deliver actions that system components must perform. When time restrictions are present, the system is said to be real-time if at least one of the tasks is performed but it must be executed before a certain deadline. XMPP and DDS protocols are designed for real-time communication. Even though the other protocols, such as MQTT and CoAP, have received a lot of attention due to their simplicity and scalability, none of them support real-time interactions. To address this, many approaches are proposed to add enhancement to applications protocols without changing their simplicity and scalability. Kim et al. [31] propose to integrate MMS and MQTT protocol for Internet of Things industrial applications. Konieczek et al. [32] presented a lightweight Java implementation of the Constrained Application Protocol called jCoAP that enables CoAP-based communication for embedded devices with comparably small latencies (real-time interaction).

IoT devices are constrained. They have limited capabilities, memory, and energy. And the use of heavy communication protocols on these devices reduces the performance of IoT communication. i.e shut down the devices quickly, increase the delay of communication. Power consumption is one of the most constrained aspects of IoT devices, "which makes the most powerful applications protocols not suitable for Internet of Things ecosystems. Although MQTT is a lightweight protocol, it has its drawbacks for extreme environments. MQTT clients must support TCP and would normally keep an open connection to the broker at all times where packets loss and connection drop rates are high or computing resources are scarce. Moreover, topic names are often long ones which make the header bigger and use significant bandwidth and power as well. To address this, many variations and enhancements are proposed. First, Query Telemetry Transport for Sensor Networks (MQTT-SN) was created [33], which runs over UDP. UDP is mainly used for sensor nodes and devices with low computing performance. MQTT-SN requires additional gateways to connect the clients to the MQTT broker over UDP which can be suitable for devices with multicast support. A modification to MQTT-SN with additional security elements adopted from DTLS is proposed to replace the DTLS protocol to enable shorter lightweight packet headers [7]. Akintade et al. [34] proposed another architecture to facilitate the development of energy-efficient and low-cost IoT solutions, namely, the aMQTT architecture. The architecture is based on the existing MQTT architecture and the low cost ESP8266 IoT hardware platform. Second, many enhancement solutions were added to the MQTT-SN protocol to increase its performance especially in extremely lossy channels where

re-transmission creates a huge overhead in terms of power consumption, delay, and processing. Alshantout et al. [35] created MQTT-SN with LT (MQTT-SN-LT). They aim to use Luby Transform Codes (LT) with the MQTT-SN-QoS1 protocol without changing the protocol itself. The authors [36] proposed to add Network Coding to an MQTT-SN network.

Application protocols were not designed with security in mind [37]. They are based on common security solutions, such as DTLS and TLS which are not sufficient for optimal security as they reduce the performance of IoT systems. To go further, these solutions are very heavy for constrained devices. Added to that, certain attacks are no longer covered by these solutions which require the development of new standards to improve the security levels of each protocol. In the rest of this section, several attacks and problems are cited as well as their solutions. The flow of the distributed messages between the users of application protocol based on Publisher-Subscriber models is insecure. Wherein authentication layer authenticating credentials are sent in plain text and some form of encryption should be used. In authorization layer all users connected to the broker are listening to a Topic and receiving all the information. For authentication layer, ChaCha20-Poly1305 AEAD solution is proposed as a lightweight security scheme for MQTT/MQTT-SN communication in [38]. Since, in a MQTT environment, a user in the broker's access is authorized to access all information, after their connection to the broker the user is listening to a Topic and receiving all the information. A new solution of certified authority is opted for in [39] to generate two kinds of certificates, the first one for the client and the second one for the Topics. Denial-of-sleep attacks aim at depriving victims of devices entering low-power sleep mode. Since the CoAP protocol suffers from this type of attack where Internet-located attackers can force IoT devices that run CoAP servers to expend much energy by sending lots of CoAP messages to them, a new solution is proposed by adding a block to filter the CoAP messages en route before entering the network [40].

Interoperability is meant to make communication among heterogeneous devices and software applications from different vendors possible. Interoperability has four dimensions: technical, syntactical, semantic, and organizational interoperability [41]. There is no compatibility in intercommunication between application protocols. Messages are not supposed to be exchanged. Thus, we need new standards to convert communication protocols and to enlarge the protocol's capabilities for larger interoperability. A new efficient application layer gateway that converts MQTT messages into HTTP is proposed in [42]. To address the problems of the interconnection of embedded systems in networks, the authors of [43] aim to dynamically model and create links between MQTT brokers based on multi-agent systems to establish the highest level of connectivity for brokers to ensure maximum transmission of messages to subscribing clients. Since there is no compatibility between the sensors, where each sensor for example has its own data display units, there is a need for common semantics for these sensors. To solve this prob-

lem, several standards have been developed to ensure that the precise meaning of exchanged information can be understood by any other application that was not initially developed for that purpose. A semantic data extraction implementation over MQTT for Internet of Things centric wireless sensor networks was introduced [44].

The Quality of Service (QoS) characterizes the quality of communication links between nodes. Generally, it is the capacity to carry the traffic between nodes in the best condition, such as in terms of availability, packet loss rate, and throughput. So, to ensure good communication it is recommended to define clearly the quality metrics and to enhance the communication protocols accordingly. Quality of Service is the strength of application layer protocol, that represents the ability to configure the performance and reliability of the network. Some protocols do not define any QoS level which reduces their performance, while others, such as CoAP, MQTT, and DDS define different levels of QoS which address different requirements, such as message delivery, timing, loose coupling, and fault tolerance. As MQTT provides only three levels of QoS for different classes of traffic, so many drawbacks arise. Firstly, the traffic flow between subscribers and publishers is not controlled since publishers send data to broker and broker forwards it to subscribers which could increase the number of packet losses and delays. A new flow control mechanism is designed to overcome the flow control problem of MQTT where the publisher can overwhelm the subscriber [45], [46]. Secondly, MQTT does not support the urgency of the message. Hence, normal and urgent messages are processed with the same priority. Many approaches are designed for this purpose. Hwang et al. [47] proposed a new method to expand the functions of the MQTT to transmit urgent messages first by creating a U-Mosquitto broker capable of processing urgent messages. However, MQTT protocol has vulnerability to maintain order between messages, which is very important in some home automation, such as controlling gas valve. Hwang et al. [48] designed and implemented a reliable message transmission system using MQTT protocol to maintain messages order. The absence of a standard for controlling the number of messages received is such a serious problem where the subscriber devices are forced to receive all messages even if they do not need to receive them frequently. To solve this problem, reducing the delivery of unnecessary messages is the best solution. Hwang et al. [49] focused on the MQTT protocol that is currently used to deliver messages between IoT devices and proposed the concept of Reception Frequency Control (RFC), which is designed to control the frequency at which subscribers receive messages.

VI. CONCLUSIONS

Application communication protocols in IoT ecosystems are used to successfully interact between IoT devices and servers / Clouds that process the information collected. Application protocols specific to IoT have been developed to meet the requirements of devices with limited resources, and those of networks with low bandwidth and high latency. However, es-

TABLE I
SURVEYS ON COMMUNICATION PROTOCOLS CHALLENGES AND ENHANCEMENT.

Challenge	Focus	Protocol	References
Real-time communication	Industrial application IoT based system Embedded devices Prototype Medical Instruments Applied to Neurodegenerative Disease Diagnosis	MQTT MQTT CoAP MQTT/AMQP	[31] [50] [32] [51]
Constrained devices	Power saving Power saving Power saving Decrease the computational complexity of the clients	MQTT MQTT-SN CoAP MQTT	[33], [34] [35], [36] [52] [53]
Interoperability	Technical Interoperability Technical Interoperability Syntactical interoperability Semantic interoperability	MQTT/HTTP MQTT All protocols MQTT	[42] [43] [54] [44]
Security	Authentication User authority to information access User Registration Denial-of-sleep attacks	MQTT/MQTT-SN MQTT MQTT CoAP	[38], [55] [39] [56] [40]
Quality of services	Control the traffic flow between the subscribers and publishers Maintain message order Transit urgent message first Reduce the delivery of unnecessary messages Data Delivery in Mobile Scenarios Network Congestion Control Object Discovery	MQTT MQTT MQTT MQTT MQTT CoAP CoAP	[45], [46] [48] [47] [49] [57] [58] [58]

tablishing low-cost communications is not enough. These protocols must allow data to be exchanged and this data must be understood by the entities of different types which receive them. The interoperability of distributed applications is defined as the ability of success for the IoT thanks to a set of application protocols for users to communicate and exchange data and services, wherever they are in the world regardless of the origin of the equipment they use.

The application protocols are still quite young and have yet to demonstrate their reliability. However, several challenges remain to be taken up to further broaden their acceptances. These challenges are related to the drawbacks of application protocols. The limits of these protocols have an important size. They can be summarized in the following points: not suitable for real-time and industrial application, not suitable for constrained devices and lack of interoperability, security mechanisms and Quality of Service (QoS).

In this paper, we surveyed the most suitable communication protocols for the Internet of Things and related challenges of IoT issues by introducing relevant and recent approaches for improving the performance of application layer IoT systems.

The studied application protocols, in general, are based on MQTT, CoAP applications protocols. This is justified due to MQTT and CoAP being already the most suitable solutions in IoT since they are initially better attending to the application layer criteria: message size, overhead, power consumption, resource requirement, bandwidth, and reliability.

bility.

REFERENCES

- [1] IRENA, "Internet of things innovation landscape brief," International Renewable Energy Agency, Abu Dhabi, 2019.
- [2] M. Atiqzaman, M. Noura, and M. Gaedke, "Interoperability in internet of things: Taxonomies and open challenges," *Mobile Networks and Applications*, pp. 796–809, 07 2018.
- [3] G. Salazar Chacón, C. Venegas, M. Baca, I. Rodriguez, and L. Marrone, "Open middleware proposal for iot focused on industry 4.0," 2018 IEEE 2nd Colombian Conference on Robotics and Automation (CCRA), pp. 1–6, 11 2018.
- [4] S. Nuratch, "Applying the mqtt protocol on embedded system for smart sensors/actuators and iot applications," 2018 15th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, no. 1, 2018.
- [5] M. Iglesias Urkia, A. Orive, and A. Urbieto, "Analysis of coap implementations for industrial internet of things: A survey," *Procedia Computer Science*, pp. 188–195, 12 2017.
- [6] M. Mónica, G.-R. Carlos, and C. Celeste, "Performance evaluation of coap and mqtt-sn in an iot environment," *Proceedings*, vol. 31, p. 49, 11 2019.
- [7] S. Jaikar and R. Kamatchi, "A survey of messaging protocols for iot systems," *International Journal of Advanced in Management, Technology and Engineering Sciences*, 02.
- [8] J. Dizdarevic, F. Carpio, A. Jukan, and X. Masip, "A survey of communication protocols for internet of things and related challenges of fog and cloud computing integration," *ACM Computing Surveys*, 04 2018.
- [9] T. Sultana, "Choice of application layer protocols for next generation video surveillance using internet of video things," *IEEE Access*, pp. 41607 – 41624, 2019.
- [10] Z. B. Babovic, J. Protic, and V. Milutinovic, "Web performance evaluation for internet of things applications," *IEEE Access*, 2016.

- [11] T. Perumal, S. K. Datta, and C. Bonnet, "Iot device management framework for smart home scenarios," in 2015 IEEE 4th Global Conference on Consumer Electronics (GCCE), pp. 54–55, 2015.
- [12] A. Ahmed, "Secured service discovery technique in iot," *Journal of Communications*, vol. 14, pp. 40–46, 01 2019.
- [13] N. Naik and P. Jenkins, "Web protocols and challenges of web latency in the web of things," 2016 Eighth International Conference on Ubiquitous and Future Networks (ICUFN), pp. 845–850, aug 2016.
- [14] N. Naik, "Choice of effective messaging protocols for iot systems: Mqtt, coap, amqp and http," 2017 IEEE International Systems Engineering Symposium (ISSE), pp. 1–7, 2017.
- [15] A. Esfahani and et al., "A lightweight authentication mechanism for m2m communications in industrial iot environment," *IEEE Internet of Things Journal*, pp. 1–1, 08 2017.
- [16] D. Soni and A. Makwana, "A survey on mqtt: A protocol of internet of things(iot).," *International Conference on Telecommunication, Power Analysis And Computing Technique (ICTPACT)*, 2017.
- [17] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: A survey on enabling technologies, protocols, and applications," *IEEE Communications Surveys Tutorials*, no. 4, pp. 2347–2376, 2015.
- [18] A. Hornsby and R. Walsh, "From instant messaging to cloud computing, an xmpp review," *Proceedings of the International Symposium on Consumer Electronics, ISCE*, pp. 1 – 6, 07 2010.
- [19] Z. B. Babovic, J. Protic, and V. Milutinovic, "Web performance evaluation for internet of things applications," *IEEE Access*, pp. 6974–6992, 2016.
- [20] C. A. L. Putera and F. J. Lin, "Incorporating oma lightweight m2m protocol in iot/m2m standard architecture," 2015 IEEE 2nd World Forum on Internet of Things (WF-IoT), pp. 559–564, 2015.
- [21] OMA, "Oma device management protocol approved version 1.2.1," 06 2008.
- [22] OMA, "Oma device management protocol approved version 1.3," 05 2016.
- [23] OMA, "Device management architecture candidate version 1.3," 03 2012.
- [24] O. M. Alliance, "Lightweight machine to machine technical specification," 02 2018.
- [25] G. Association, "Iot standard diagnostic logging version 1.0," official document," 01 2018.
- [26] F. Broadband, "Tr-069 cpe wan management protocol," 03 2018.
- [27] F. Broadband, "Tr-069 cpe wan management protocol v1.1," 09 2007.
- [28] G. Nebbione and M. Calzarossa, "Security of iot application layer protocols: Challenges and findings," *Future Internet*, p. 55, 03 2020.
- [29] C. Cabrera, A. Palade, and S. Clarke, "An evaluation of service discovery protocols in the internet of things," *Proceedings of the Symposium on Applied Computing*, p. 469–476, 2017.
- [30] A. Bröring, S. K. Datta, and C. Bonnet, "A categorization of discovery technologies for the internet of things," *IoT'16*, 2016.
- [31] D. Kim, H. Lee, and D. Kim, "Enhanced industrial message protocol for real-time iot platform," 2018 International Conference on Electronics, Information, and Communication (ICEIC), pp. 1–2, 2018.
- [32] B. Konieczek, M. Rethfeldt, F. Golasowski, and D. Timmermann, "Real-time communication for the internet of things using jcoap," 2015 IEEE 18th International Symposium on Real-Time Distributed Computing, pp. 134–141, 2015.
- [33] U. Hunkeler, H. L. Truong, and A. Stanford-Clark, "Mqtt-s — a publish/subscribe protocol for wireless sensor networks," 2008 3rd International Conference on Communication Systems Software and Middleware and Workshops (COMSWARE '08), pp. 791–798, 2008.
- [34] O. Akintade, T. Yesufu, and L. Kehinde, "Development of an mqtt-based iot architecture for energy-efficient and low-cost applications," *International Journal of Internet of Things*, pp. 27–35, 06 2019.
- [35] A. Alshantout and L. Al-Awami, "Enhancing mqtt-sn performance via fountain codes in extreme conditions," 2019 15th International Wireless Communications Mobile Computing Conference (IWCMC), pp. 1184–1189, 2019.
- [36] B. Schütz, J. Bauer, and N. Aschenbruck, "Improving energy efficiency of mqtt-sn in lossy environments using seed-based network coding," 2017 IEEE 42nd Conference on Local Computer Networks (LCN), pp. 286–293, 2017.
- [37] A. Oak and R. D. Daruwala, "Assessment of message queue telemetry and transport (mqtt) protocol with symmetric encryption," 2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC), pp. 5–8, 2018.
- [38] O. Sadio, I. Ngom, and C. Lishou, "Lightweight security scheme for mqtt/ mqtt-sn protocol," 2019 Sixth International Conference on Internet of Things: Systems, Management and Security (IOTSMS), pp. 119–123, 2019.
- [39] A. Mektoubi, H. L. Hassani, H. Belhadaoui, M. Rifi, and A. Zakari, "New approach for securing communication over mqtt protocol a comparison between rsa and elliptic curve," 2016 Third International Conference on Systems of Collaboration (SysCo), pp. 1–6, 2016.
- [40] F. Seidel, K.-F. Krentz, and C. Meinel, "Deep en-route filtering of constrained application protocol (coap) messages on 6lowpan border routers," 04 2019.
- [41] T. Vadluri, "A research on interoperability issues in internet of things at application layer," *Mobile Netw Appl*, 03 2020.
- [42] M. A. A. da Cruz, J. J. P. C. Rodrigues, E. S. Paradello, P. Lorenz, P. Solic, and V. H. C. Albuquerque, "A proposal for bridging the message queuing telemetry transport protocol to http on iot solutions," 2018 3rd International Conference on Smart and Sustainable Technologies (SpliTech), pp. 1–5, 2018.
- [43] S. Alexandre, F. Carlier, and V. Renault, "Dynamic bridge generation for iot data exchange via the mqtt protocol," *Procedia Computer Science*, pp. 90–97, 01 2018.
- [44] S. Wagle, "Semantic data extraction over mqtt for iot-centric wireless sensor networks," 2016 International Conference on Internet of Things and Applications (IOTA), pp. 227–232, 2016.
- [45] A. S. Sadeq, R. Hassan, S. S. Al-rawi, A. M. Jubair, and A. H. M. Aman, "A qos approach for internet of things (iot) environment using mqtt protocol," 2019 International Conference on Cybersecurity (ICoCSec), pp. 59–63, 2019.
- [46] A. Sadeq, R. Hassan, and A. Jubair, "Enhanced mqtt for providing qos in internet of things (iot): A study," *Advanced Science Letters*, pp. 5199–5203, 07 2018.
- [47] K. Hwang, J. M. Lee, I. H. Jung, and D. Lee, "Modification of mosquito broker for delivery of urgent mqtt message," 2019 IEEE Eurasia Conference on IOT, Communication and Engineering (ECICE), pp. 166–167, 2019.
- [48] H. Hwang, J. Park, B. Lee, and J. Shon, "An enhanced reliable message transmission system based on mqtt protocol in iot environment," *Lecture Notes in Electrical Engineering*, pp. 982–987, 01 2017.
- [49] K. Hwang, J. Moon Lee, and I. Hwan Jung, "Extension of reception frequency control to mqtt protocol," *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, pp. 2278–3075, 06 2019.
- [50] J. Finochietto, G. Eggly, R. Santos, J. Orozco, S. Ochoa, and R. Meseguer, "A role-based software architecture to support mobile service computing in iot scenarios," *Sensors*, p. 4801, 11 2019.
- [51] A. Depari, D. Carvalho, P. Bellagente, P. Ferrari, E. Sisinni, and A. Padovani, "An iot based architecture for enhancing the effectiveness of prototype medical instruments applied to neurodegenerative disease diagnosis," *Sensors*, vol. 19, p. 1564, 03 2019.
- [52] W. Mardini, M. Bani Yassein, M. N. Alrashdan, A. Alsmadi, and A. Bani Amer, "Application-based power saving approach for iot coap protocol," *Proceedings of the First International Conference on Data Science, E-Learning and Information Systems*, 10 2018.
- [53] R. Giambona, A. Redondi, and M. Cesana, "Mqtt+: Enhanced syntax and broker functionalities for data filtering, processing and aggregation," *ArXiv*, 10 2018.
- [54] M. Al-Osta, B. Ahmed, and A. Gherbi, "A lightweight semantic web-based approach for data annotation on iot gateways," *Procedia Computer Science*, pp. 186–193, 12 2017.
- [55] C. Fetzter, G. Pfeifer, and T. Jim, "Enhancing dns security using the ssl trust infrastructure," pp. 21– 27, 03 2005.
- [56] S. Jakeer Hussain and G. Ramyarsi, "Security enhancement for devices using restapi, middleware iot gateway," *International Journal of Engineering and Advanced Technology (IJEAT)*, p. 2249 – 8958, 09 2019.
- [57] J. Luzuriaga, M. Perez, P. Boronot, J.-C. Cano, C. Calafate, and P. Manzoni, "Improving mqtt data delivery in mobile scenarios: Results from a realistic testbed," *Mobile Information Systems*, pp. 1–11, 07 2016.
- [58] M. Swarna and T. Godhvari, "Enhancement of coap based congestion control in iot network - a novel approach," *Materials Today: Proceedings*, 2020.