Received November 24, 2019, accepted December 11, 2019, date of publication December 18, 2019,
date of current version January 7, 2020.

Digital Object Identifier 10.1109/ACCESS.2019.2960508

An Intelligent Algorithm for Resource Sharing and Self-Management of Wireless-IoT-Gateway

PEDRO LUIS GONZÁLEZ RAMÍREZ[1,3], MIRAN TAHA[2], (Member, IEEE), JAIME LLORET[3], (Senior Member, IEEE), AND JESÚS TOMÁS[3]

[1]Departamento de Ingeniería Electrónica, Universidad Central, Bogotá 110311, Colombia
[2]Department of Computer Science, University of Sulaymaniyah, Sulaymaniyah 46001, Iraq
[3]Instituto de Investigación para la Gestión Integrada de zonas Costeras, Universitat Politècnica de València, 46730 Valencia, Spain

Corresponding authors: Pedro Luis González Ramírez (pgonzalezr1@ucentral.edu.co), Miran Taha (miran.abdullah@univsul.edu.iq), Jaime Lloret (jlloret@dcom.upv.es), and Jesús Tomás (jtomas@upv.es)

This work was supported in part by the "Ministerio de Economía y Competitividad", through the "Convocatoria 2014 Proyectos I+D - Programa Estatal de Investigación Científica y Técnica de Excelencia" in the "Subprograma Estatal de Generación de Conocimiento", under Grant TIN2014-57991-C3-1-P and through the "Convocatoria 2017 - Proyectos I+D+I - Programa Estatal de Investigación, Desarrollo e Innovación, convocatoria excelencia" under Grant TIN2017-84802-C2-1-P.

ABSTRACT Internet of Things (IoT) is rapidly gaining momentum in the scenario of telecommunications. Conventional networks allow for interactivity and data exchange, but these networks have not been designed for the new features and functions of IoT devices. In this paper, an algorithm is proposed to share common recourse among Things, that is, between different types of smart appliances. This proposal is based on an IoT network with centralized management architecture, controlled by an Artificial Intelligence (AI). The AI controller uses an algorithm which based on machine learning techniques, collecting information on the network through an information protocol. Every smart thing that connects to the network is announces through a protocol message called Function and Service Discovery Protocol (DFSP) over the queued message telemetry transport protocol (MQTT). The proposed algorithm is responsible for discovering and allocating resources in the networks. As a result, using our proposed algorithm in communication system provides the outperform efficiency and availability than that used in conventional communication systems for the integrate IoT devices.

INDEX TERMS Internet of Things, artificial intelligence, resources sharing, IoT gateway, collaborative groups, architecture with centralized management, MQTT protocol.

I. INTRODUCTION

The Internet of Things (IoTs) is a novel paradigm of integration of several technologies that is rapidly gaining momentum in the scenario of telecommunications, especially in wireless network [1]. In traditional communications systems, usually within a house or an office several personal computers and smart devices are seamlessly connected together through a wireless router. In this network, resources can be shared; the most common example is a printer and access to the Internet.

Therefore, the integrated intermediary network devices, such as switches and routers, are used to interconnect the network computers in order to exchange data. These devices manage and control the Internet access with different network policies. Moreover, in IoT scenarios, heterogeneous things use the seamless intermediary network devices to share and exchange data, which is recognized as IoT gateway that are often used between sensor networks and the Internet to provide advanced services.

As a consequence, the IoT gateway that once had a specific function more towards sensor networks, can now provide information on the flow of data that passes through it in both directions, acting as a translator. This allows two different protocols to communicate and route data to the Internet [2]. However, the reprogramming of its functions is limited to the few options that the manufacturer allows in its configuration. This reprogramming can be done by the user directly in a configuration panel or by software from a mobile application.

The associate editor coordinating the review of this manuscript and approving it for publication was Sherali Zeadally.

The issue currently with an IoT gateway, is that not smartly self-managed or allowed to take efficient control of the exchange of resources between heterogeneous things of an IoT network.

Although, there is a variety of research about IoT Gateway for smart devices. A novel Machine-to-Machine (M2M) service is introduced in [3]. They proposed self-configurable gateway and configuration of smart things over the wireless networks. As well, an IoT gateway is designed and implemented through creating a unified connection to the technological layer aided by IoT in [4]. This proposed gateway allows management of devices and identification of new device by creating virtual representations of the physical devices. An auto configuration solution based on interpretable configuration is presented in [5], which focused on some algorithms for computing the IoT gateway configurations. Five algorithms are contributed, while a thorough evaluation reveals which of the algorithms should be used in different operation scenarios in order to achieve high fulfillment of the operator's target. Therefore, a case study is provided in [6], to better known how to design and implement IoT-gateway for home environment, which is based on suitable self-configuration and scalability.

In this paper, we propose a new algorithm for allocating and sharing the common resources among the heterogeneous devices in the scenarios of IoT such as smart cities, smart grids, industrial automation, smart driving, elderly assistance, or home automation, and others.

The proposal is based on an IoT network with centralized management architecture, controlled by an Artificial Intelligence (AI). The AI controller is an algorithm based on machine learning techniques, collecting information on the network through an information protocol. Every smart thing that connects to the IoT network deployed message through Protocol of Discovery of Functions and Services (DFSP). The information of the available and common resources is shared among the things in the integrated network.

The rest of the paper is structured as follows. Related work with IoT Gateways resources sharing in smart cities, controlled by artificial intelligence, will be explained in section 2. Then, in Section 3, we state the problem and describe the resource allocation proposal method for wireless IoT network. Section 4 presents the performance of the tests and results' discussion. Finally, in Section 5, we highlight our conclusion and future work.

II. RELATED WORK

Internet of things is growing very fast and IoT gateway becomes an important part in its structure. IoT gateway allows supporting a variety of communication protocols and data exchange between various IoT-nodes, The goal of IoT gateway is to bridge various sensing domain networks with public communication networks or Internet, settle with the heterogeneity between these various networks, strengthen the management of both IoT gateway itself and terminal nodes [3]. Generally, García-García *et al.* [1] presented an overview of the premise of the emerging of the Internet of Things (IoT) according to enabling technologies, protocols, and applications. They provided a good foundation to gain an insight into the IoT technologies and protocols to understand the overall architecture and role of the different components and protocols that constitute the IoT.

Zhong *et al.* [7] outlined the disadvantages of the practical application with three-layer architecture of IoT. In order to better interpret the meaning and features of the IOT, and discusses the gateway technology which connecting the sensing network and traditional communication network, layers system architecture is explained. They designed IoT application scheme with using the IoT Gateway as a bridge in order to realize exchanges of the information and communication of different equipment in the industry services. Therefore, the sensors and devices use numerous protocols and communication methods for exchanging the data. In a traditional setup, it is difficult to connect multiple devices with different proprietary applications and monitor. Moreover, it is heavy to control multiple devices and store, analyses all the vitals parameter data together. There is no common platform available to connect the multiple sensors and to monitor the vitals simultaneously. Selvaraj and Kalambettu [8] proposed the comprehensive, scalable, plug and play smart gateway for connecting variety of health sensors and allows the smart devices seamless communication for transmit and control the resources. With IoT-Gateway enabled the framework seamlessly connects with cloud services and stream the data for storage, analysis and prediction. The proposed unified framework is also enable seamless connectivity between gateway and cloud platform and services.

The architecture of the heterogeneous Internet of Things (IoT) gateway and the problems of its integration with the Industrial Internet of Things (IIoT) are investigated in [9] by Viacheslav Et al. the issue of the usage of heterogeneous gateway for conversion of packages raised when various protocols are formed in IoT. Therefore, they proposed Industrial Internet of Things Conversion Format (IICF) for converting various protocols of the Industrial Internet of Things among themselves. The proposal based on the analyzed architecture of the heterogeneous IoT gateway, structure of interaction software system, structure of the model network for interoperability different protocols. Thus, Rodrigues *et al.* in [10] proposed a novel IoT-based mobile gateway solution for mobile health scenarios. The gateway autonomously collects information about the user/patient location, heart rate, and possible fall detection. Moreover, it forwards the collected information to a caretaker IPA, in real timet hat manage a set of actions and alarms appropriately. The algorithms used for each mobile gateway service, and the scenarios where the mobile gateway acts as a communication channel or a smart objects. Also, Athreya *et al.* [11] took a preliminary look at software, hardware, and network architectures involved in IoT systems. They summarize that the various network components to operate and interoperate effectively the smart devices must be able to coordinate their management capabilities.

They proposed the use of self-management and self-adaptation to cope with countless dynamics. The proposal presents an underlying framework for self-managing devices, comprising measurement based learning and adaptation to changing system context and application demands.

Kanchana and Susanth [12] explained that the gateways are not affordable in case of simple IoT lab experimentations because things communicate in different networks. An IoT gateway, which provides device connectivity and protocol translation, is often expensive in case of small-scale academic projects. A method is provided to emulate the functionality of an IoT gateway using a local computer instead of a separate hardware. The proposed EmI gateway is successfully tested to provide communication between sensor nodes in heterogeneous networks such as GPRS, WiFi, Ethernet, RF, and Bluetooth and Amazon's cloud server. The EmI gateway is used in the IoT based application of sewage flow scheduling that involves sensor nodes connected in different communication networks.

Therefore, the studies are presented above on IoT gateways do not offer dynamic resource allocation information protocols for that things can use it. Therefore, it is necessary to explore proposals that allow for different ways of assigning resources and sharing dynamically among the smart devices in the networks. Especially when the current proposals already speak on IoT networks with AI intervention and M2M techniques. For this reason, we propose IoT-gateway algorithm based on artificial intelligent to allow dynamic self-management of gateways when the resources needed to be assigned and shared between IoT-nodes in wireless networks.

III. PROPOSAL DESCRIPTION

A. PROBLEM

The current data networks allow to connect "Things", but under their policies and protocols. Although protocols such as transmission control (TCP/IP) are flexible and allow transport to other protocols in the network, it is a low level protocol, which does not solve all the problems that appear in an IoT network [13]. Things must be managed differently and the resources that Things need are not the same as a conventional network.

Some IoT protocols such as MQTT can exchange messages on these types of networks, as well as the Constrained Application Protocol (CoAP) and Representative State Transfer (REST) [14]. MQTT and CoAP are M2M protocol and REST uses HTTP to perform operations between client and server.

However, the operation of these protocols is not designed to share, use or allocate resources on the network.

For that, is necessary a protocol or autonomous entity that allow to Things talk between them without human intervention, only machines. In this case, it has been decided to work with MQTT protocol [15], since is open source and allow modifying its operating mechanics. MQTT performs the M2M communication through a central server (MQTT Broker) inside a Gateway or directly through the cloud (Cloud Broker). This is a publication/subscribe protocol and was initially aimed at IoT sensors networks [16], since its main target was to optimize bandwidth and minimize the hardware and the processing [1].

However, is possible that when creating an adaptive algorithm, it is can take advantage of its own messages to send over they the new messages of the DFSP protocol designed for this proposal.

Another problem is that wireless routers in home networks are not flexible and do not allow reprogramming of protocols. Commercially this type of devices are obtained, but with very limited functions. To achieve modifications in the network and that Things operate under another type of architecture, it is necessary to change the traditional router to replace it with a reprogrammable one.

In addition, Things do not have the capacity to process information because their factory functions are limited and they cannot collaborate and interact with other things in a network. It is not enough to have a means of communication to the internet, it is necessary to increase the processing capacity.

From the above, it is evident that the devices that make up a conventional data network, such as the host, the router and the cloud platform, must be reconsidered.

B. PROPOSAL

The next proposal consists of creating three control agents of the same type but in different environments. This control agent is the AI, which will be inside the smart things, in the Gateway with centralized administration and the platform in the cloud. The AI is based on the same principle and has two main functions, managing the algorithms based on Machine Learning techniques and learning the relations Machine to Machine (M2M) as the product of the joint work of a group of Things.

It is necessary to see the problem first since the perspective of the "Things", because these just work when its user controls it or if before it had a previous programming. For that, the things carry its own control, it is necessary to know that resource, function or service is shared between the things.

This paper addresses the study of the problem through the analysis of the exchange of DFSP protocol messages between smarts things, the gateway, and the cloud platform. In this way, it is can see how each Thing that connects announces its functions and services and shares it on the network.

On the other hand, the AI that administers the algorithm that assigns the role to each Thing decides that Thing is a resource or makes available a resource. This is an adaptive algorithm and changes according to the role that each Thing assumes within the group work.

C. NETWORK IOT ARCHITECTURE WITH CENTRALIZED MANAGEMENT

To understand the architecture that will use in this proposal is fundamental to explain it through of three actors and

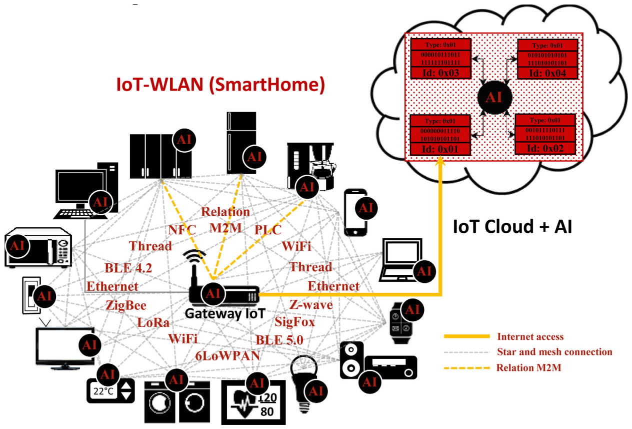


FIGURE 1. Network IoT-WLAN architecture.

its relation: Smart Things, the IoT Gateway and the IoT platform, each contain an AI.

Through the communication of the AI between these three devices, it is possible to achieve a harmony of collective learning in the entire network.

The design of this architecture is thought for a Wireless Local Area Network (WLAN), connecting Things with different interconnection technologies and IoT protocols, depending on their use, bandwidth, processing capacity, and distance [17]. The idea is to modify a network home conventional and become it in a WLAN for IoT. The Figure 1, show an example of network IoT-WLAN, with different topologies and IoT protocols.

The connection of the Gateway to the intelligent platform in the cloud is done through the Internet and constantly monitors what happens in the house. If it is realized queries to the cloud, the platform validates the type of request, classifies it and decides if it is necessary to connect to other platforms else the AI into gateway decides if the problem can be resolved on the local network through the M2M connections [18].

Smart Things + AI: The importance of things acting intelligently, not just improves its internal functions, but it also allows defining functions and services that put to disposition to other things in the network.

Things, in this case study will be appliances inside a house. These require the ability to process and communicate with different communication technologies.

In the architecture of Figure 1, it is can see how everything connects directly to other things or through the Gateway. This connection type will depend on the permissions and relations established by the Gateway.

For example, yellow line of figure 1 depicts the relation M2M established for the gateway where creating a group with something in common.

IoT Gateway + AI: This multiprotocol device allows managing and centralizing all of the information regardless of the underlying technology of interconnecting. It carries out the management in a centralized way in the network and allows all the Things in the network to connect with each other, initially passing through it. In this way, the M2M relation are only made at the gateway level, that is, the central server (MQTT Broker) is not in the cloud.

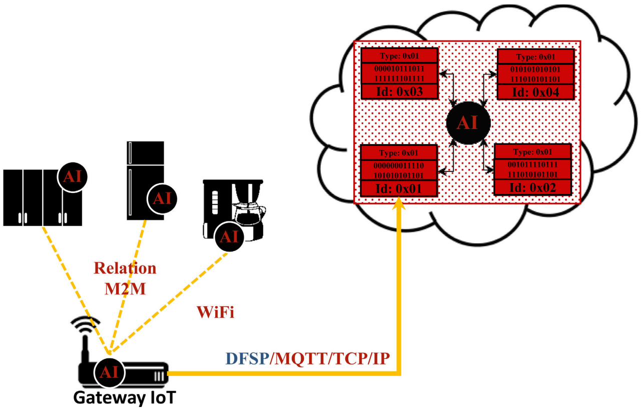


FIGURE 2. Allocation of resources by group of things.

For this reason, the AI learns about the relation that are commonly established between Things when a group of Things works together. Another indispensable resource is to manage access to the internet. The AI will decide whether the packets are forwarded within the same network or outside the network, based on the relation, the resolution of a problem and a clear reason for the requests to be sent out of the network to an IoT platform in the cloud.

IoT Platform + IA: Its function is to maintain network monitoring through the gateway and deliver information only when the gateway requests it.

With the information consulted to Internet through of platform in the cloud [17], the Things on the network can complement the local information and make a better decision.

Example: Using the architecture of Figure 1, all things are connected to each other through the Gateway (MQTT Broker) + AI. Only the MQTT protocol will be used to establish M2M relations over WiFi connections. The AI that resides on the Gateway reviews and analyzes the payload of each MQTT packet continuously and creates statistics for each request. When it detects that there are continuous requests for common activities between the machines, it creates logical work groups. In Figure 2, these three things highlighted in Figure 1 are isolated to explain their relation in more detail. In this case, it has not yet been established who or what resource is needed. This assignment, is the job that the adaptive algorithm of resource allocation controlled by the AI must perform, see Figure 3.

Figure 2 shows a group of Things doing a joint work. The M2M relation that the AI establishes depends on the functions and the service provided by the Things. For example, it could be assumed that something in common between this group of Things is the preparation of Coffee as a service to the User and each Thing can perform a function that helps with the provision of this service. Coffee is a resource that has the kitchen cupboard and milk is a resource that has the refrigerator. The AI activates the algorithm that is responsible for the allocation of resources and the DFSP protocol collects the information.

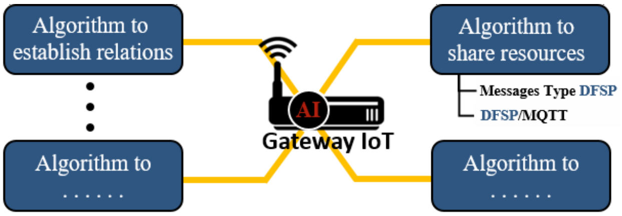


FIGURE 3. IA controlling the algorithms.

In Figure 3, the IoT gateway controls different types of algorithms through the AI and learns from them through the training matrix they provide. Each training matrix is a data file with information about the usage habits and how a machine has been used. This provides information to the system about what kind of operations the user usually performs in front of the machine.

Each petition or service is registered and then analyzed, in order to obtain a statistical history of behavior. After analyzing it with the AI algorithm, things send the resulting matrix to the centralized AI in the gateway. In algorithm 1, is observe the general steps of the pseudocode of the control algorithm that allows the IA to delegate tasks.

Into the gateway, the AI can handle a number of algorithms and can accommodate as many as necessary and if the capacity of the machine allows it.

The algorithm "share resources" works with the information sent from the Things in the network through the MQTT payload. In the payload, another protocol has been specified that determines the type of information depending on the type of message, called DFSP. The other algorithms provide information and perform tasks in parallel to the other tasks of the network depending on the decisions made by the AI. In this case, it will only control two algorithms, the algorithm to establish relation and share resource.

Algorithm 1 Control Into IoT Gateway

```
Set an event listener(Connection request from Clients)
Initialize MQTT_Broker
Received updates on a subscribed topic
Update the information of connected Things
Activates the algorithms "establish relation" and "share resources"
Supplies initial conditions to the AI
Read training matrix of Things
Set an event listener (MQTT_Broker)
If MQTT_Broker is down then
    Disconnect from the Clients
end if
End.
```

D. AI OPERATION

As noted in the algorithm 1, the purpose of the basic algorithm AI of control in the network, is that of administering to the other algorithms that are executed in secondary and independent threads. Each algorithm is responsible for a different task but it depends on the problem that needs to be solved.

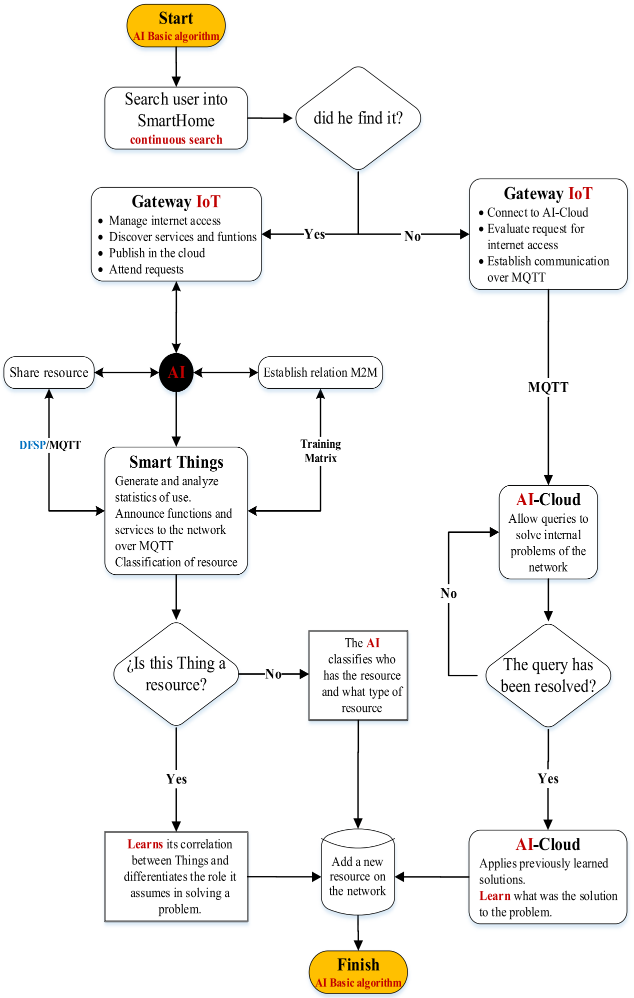

Start
AI Basic algorithm
Search user into SmartHome
continuous search
did he find it?
Yes
No
Gateway IoT
Manage internet access
Discover services and funtions
Publish in the cloud
Attend requests
Gateway IoT
Connect to AI-Cloud
Evaluate request for internet access
Establish communication over MQTT
Share resource
AI
Establish relation M2M
DFSP/MQTT
Training Matrix
MQTT
Smart Things
Generate and analyze statistics of use.
Announce functions and services to the network over MQTT
Classification of resource
AI-Cloud
Allow queries to solve internal problems of the network
No
¿Is this Thing a resource?
No
The AI classifies who has the resource and what type of resource
The query has been resolved?
Yes
Yes
Learns its correlation between Things and differentiates the role it assumes in solving a problem.
Add a new resource on the network
AI-Cloud
Applies previously learned solutions.
Learn what was the solution to the problem.
Finish
AI Basic algorithm

FIGURE 4. Flowchart of the IoT Gateway + AI.

The basic target of an intelligent network is to be able to provide better attention to the user without the need for the latter to control the machines. The idea is that the machines do so autonomously through the AI.

In Figure 4, the AI is observed administering only two algorithms and learning from the user's habits when he is at home. When a problem cannot be resolved internally, the AI decides to search the Internet for possible solutions, learns it and saves it for future problems. The circle that indicates the AI in Figure 4 has the two algorithms that it controls on the sides. Below, the AI analyzes the MQTT payload with the information from the DFSP protocol that comes from Smart Things. Both at the gateway and in smart things AI are present. However, in the flowchart has represented as only one because both have the same information exchange technique.

The next thing the AI assesses is whether the information that comes through the DFSP is a resource or is a conversation between things. If it is about a conversation that tries to solve a problem, learn it and store it. If it is about a resource then it classifies and stores it. In both cases, the iteration is direct between things, but the information passes through the gateway. Unlike other architectures, the MQTT Broker is present in the Gateway and not in the cloud. This possibly decreases network delays, since for any operation at the local level, it is not necessary to go to a server in the cloud that is far away.

To control internet access, the gateway evaluates whether the requests for things merit it. If the gateway already has, this information previously stored or because you learned it before, then answer the request without having to go to the internet. The cloud platform delivers information requested by the gateway and monitors only what the gateway allows.

E. ALGORITHM TO SHARE RESOURCES

All the algorithms that the AI manages reside in the IoT Gateway and are based on computational learning techniques (Machine Learning). This algorithm is who allows the Gateway to respond to requests and announcements of resources through the DFSP protocol. The messages of this protocol are encapsulated within the MQTT messages. Its logic is fundamentally based on establishing the role that each Thing plays and the assigning of resources.

The algorithm must know how to differentiate a function or a service from a resource depending on the joint work performed by a group of Things. From this, another algorithm controlled by the AI determines the M2M relation of a group of Things. With this information of the groups created by the AI, the algorithm can be adapted dynamically according to the service required by a User.

F. DFSP PROTOCOL

Function and Service Discovery Protocol (DFSP). This protocol was developed to be introduced within the payload of the PUBLISH messages of the MQTT protocol. That is, everything that is contained in this protocol is a string encoded in UTF.

The main function of this protocol is to discover the functions and services that announce the "Things" when connecting to the network. The algorithm that allows sharing resources in the network takes this information and decides between these two characteristics, if a Thing can be a resource or have a resource. Once the function of this protocol has been defined, the types of messages, their formatting, and the message exchange rules are presented. The header and body of the message of this protocol are simple and like the MQTT header, the size is fixed. Figure 5 shows the protocol format with an example of the ANNUNCEMENT message that transport the FUNTIONS of a thing.

Header: It has a fixed size of 1-Byte with four fields, each of 2-bits (Message Type, Data Type, Flags, RESERVED). In the red box in Figure 5, highlight the protocol header and below is the body. The format of the messages is shown in 8-bit linear datagrams per n-Bytes (8-bits x n-Bytes) one above the other.

Message Type	**Data Type**	**Flags**
0 - ANNUNCEMENT	0 - FUNTIONS	0 - Message sent
1 -DISCOVER	1 - SERVICES	1 - Message received

bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Byte 1	Message Type		Data Type		Flags		RESERVED	
	0	0	0	0	0	0	x	x
Byte 2	"H" (0x48)							
	0	1	0	0	1	0	0	0
Byte 3	"E" (0x48)							
	0	1	0	0	0	1	0	1
Byte 2	"L" (0x4C)							
	0	1	0	0	1	1	0	0
Byte 2	"L" (0x4C)							
	0	1	0	0	1	1	0	0
Byte 2	"O" (0x4F)							
	0	1	0	0	1	1	1	1
Byte 2	"," (0x2C)							
	0	0	1	0	1	1	0	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
Byte n	x	x	x	x	x	x	x	x

FIGURE 5. DFSP protocol format.

Body: The body carries the information of the type of message that has been defined in the header. This information is a string encoded in UTF, starting from the second Byte up to 250-MBytes. This information is concatenated through commas (,) to separate each of the items from functions and services from things.

Message Exchange Rules: The exchange of these messages depends on the mechanics of exchange of MQTT. The AI of each thing sends an ANNOUNCEMENT message over the MQTT PUBLISH message to the MQTT Broker.

This is a one-way message generated automatically by the AI every time a Thing is connected to the network. The identifier (Id) of the Smart Thing that was connected is found in the same PUBLISH message with the field named topic name.

On the other hand, the DISCOVER message is sent from the MQTT Broker every time a change occurs in the network or when the AI within of the IoT Gateway needs to update information.

G. M2M COMMUNICATION

Information of the start sequence of M2M communication algorithm is explained in algorithm 2. The observation of the algorithm shows that, at the beginning a listener is provided to process connecting the IoT gateway then the start sequence algorithm allows the AI in the Gateway to establish relations between the machines. When a relation is created between MQTT client and broker, the AI creates a thread of independent processing for the Things to talk to each other.

Therefore, in Figure 6 depicts the exchange of messages between a group of things through the DFSP/MQTT. The IoT-gateway is a common netowk system to share the things among the devices. It is also observed between the messages the control that one machine can have over another through an M2M protocol like MQTT. Therefore, the case is observed according to the figure, the autonomy that the machines can

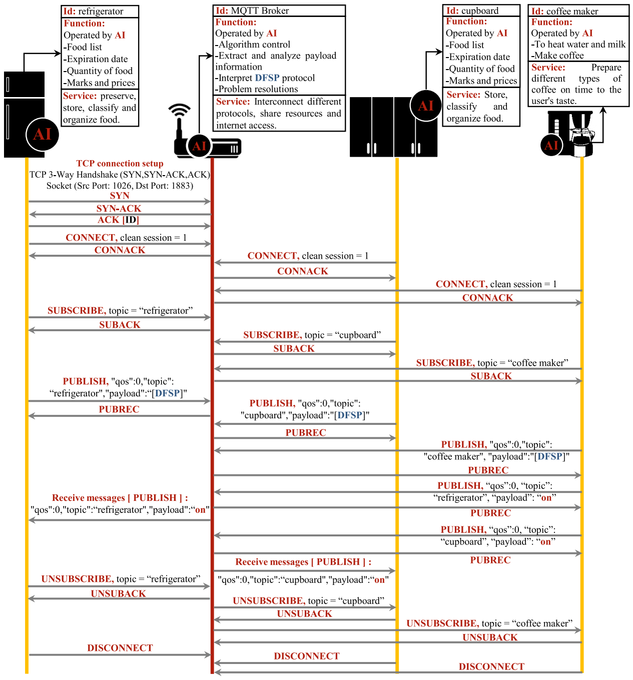

Id: refrigerator
Function:
Operated by AI
-Food list
-Expiration date
-Quantity of food
-Marks and prices
Service: preserve, store, classify and organize food.
Id: MQTT Broker
Function:
Operated by AI
-Algorithm control
-Extract and analyze payload information
-Interpret DFSP protocol
-Problem resolutions
Service: Interconnect different protocols, share resources and internet access.
Id: cupboard
Function:
Operated by AI
-Food list
-Expiration date
-Quantity of food
-Marks and prices
Service: Store, classify and organize food.
Id: coffee maker
Function:
Operated by AI
-To heat water and milk
-Make coffee
Service: Prepare different types of coffee on time to the user's taste.
AI
AI
AI
AI
TCP connection setup
TCP 3-Way Handshake (SYN,SYN-ACK,ACK)
Socket (Src Port: 1026, Dst Port: 1883)
SYN
SYN-ACK
ACK [ID]
CONNECT, clean session = 1
CONNACK
CONNECT, clean session = 1
CONNACK
CONNECT, clean session = 1
CONNACK
SUBSCRIBE, topic = "refrigerator"
SUBACK
SUBSCRIBE, topic = "cupboard"
SUBACK
SUBSCRIBE, topic = "coffee maker"
SUBACK
PUBLISH, "qos":0,"topic": "refrigerator","payload":"[DFSP]"
PUBREC
PUBLISH, "qos":0,"topic": "cupboard","payload":"[DFSP]"
PUBREC
PUBLISH, "qos":0,"topic": "coffee maker", "payload":"[DFSP]"
PUBREC
PUBLISH, "qos":0, "topic": "refrigerator", "payload": "on"
PUBREC
Receive messages [PUBLISH] : "qos":0,"topic":"refrigerator","payload":"on"
PUBLISH, "qos":0, "topic": "cupboard", "payload": "on"
PUBREC
Receive messages [PUBLISH] : "qos":0,"topic":"cupboard","payload":"on"
UNSUBSCRIBE, topic = "refrigerator"
UNSUBACK
UNSUBSCRIBE, topic = "cupboard"
UNSUBACK
UNSUBSCRIBE, topic = "coffee maker"
UNSUBACK
DISCONNECT
DISCONNECT
DISCONNECT

FIGURE 6. DFSP message exchange.

have around a common resource after the IA know through DFSP the functions and services of each Thing. It can also be observed in the exchange of the messages as dipected in the figure, the control commands on/off of the machines.

The protocol messages create a table with the functions and services of each Thing in the IoT Gateway that can be consulted by any algorithm and the AI.

Algorithm 2 M2M Relations Establishment

```
  1.
 1)  *Set an event listener* (Process connect to IoT Gateway)
 2)      The **AI** takes control and manages the Things
 3)      The **AI** initialize MQTT_Client
 4)      The **AI** connect to the MQTT_Broker
 5)      The **AI** subscribe a topic
 6)          The main topic is the name of the smart thing
 7)      The **AI** publish with the main topic
 8)          Functions and services through DFSP
 9)          Matrix training
10)      The **IA** evaluate which machines need common resources.
11)      **If** the machines are related **then**
12)              The **AI** directly controls the related machines
13)      **If** MQTT_Broker is down **then**
14)          *Disconnect* from the Clients
15)  *End.*
```

H. MQTT PUBLISH/SUBSCRIBE ARCHITECTURE

The MQTT architecture over the architecture of this proposal is simple.

The AI of each thing subscribes to the gateway (MQTT Broker) with its identifier (Id) and publishes its functions and services following the DFSP format.

In this architecture, all things are at the same level so it is not necessary to use a level separator (/).

It is could use something like /home/living-room/bulb or /home/kitchen/bulb. However, it this would only apply if all things are of the same type and therefore it is need to know their location. In this case, things are different and what matters is knowing their relation according to the work together. The syntax would be as follows: Topic/Payload/QoS. Then, the AI of each thing publishes the following structure: Id/Payload[DFSP]/0, every time it wants to talk to something else or control it.

In this structure, the cloud platform has not taken into account because communication between the gateway and the cloud have other structure.

This platform ha based on connections by group of parameters between other clouds using AI [18]. In both cases, the techniques are similar, local level the AI creates working groups and in the cloud is done by groups of parameters.

IV. PERFORMANCE TEST

In this section the performance of the test is provide. The tests of this proposal are made with the target of demonstrating that the algorithm can make decisions to assign resources depending on the information collected by the DFSP protocol.

A. IMPLEMENTATION OF THE DEVICES

The development of this proposal involves the use of single-board computers (SBC) such as the Raspberry Pi 3 Model B+ (RPi3) [19]. Each actor in this proposal, it has implemented with an RPi3, except for the cloud platform. It has tested on two different platforms for IoT: ThingsBoard [20] and ThingSpeak [21].

The idea is that each thing is controlled by an RPi3 including the IoT Gateway, under the Android Things operating system [22]. Using the Java programming language and the Android Studio development environment [23].

B. TESTBED

The technical implementation of this proposal is possible through the implementation of the RPi3, three of it emulating three Smart Things and one emulating the IoT Gateway. Another way to verify the operation of this proposal can be done through a simulation. Several simulators were studied, including Cooja [24], NS-2 [25] (Network Simulator 2), GNS3 [26] and Cisco Packet Tracer [27]. Most of the simulators are oriented to IoT sensor networks (Motes, mostly 8 and 16 bit Microcontroller based). Some of these simulators do not support the use of the RPi3 because it falls into the SBC category and cannot run a full Operating System (OS). Sometimes also, the OS of the simulators are not compatible with OS of RPi3. Currently, there is an RPi3 simulator [28], [29] that can emulate the operating system (OS-RPi3), but cannot simulate it in a network.

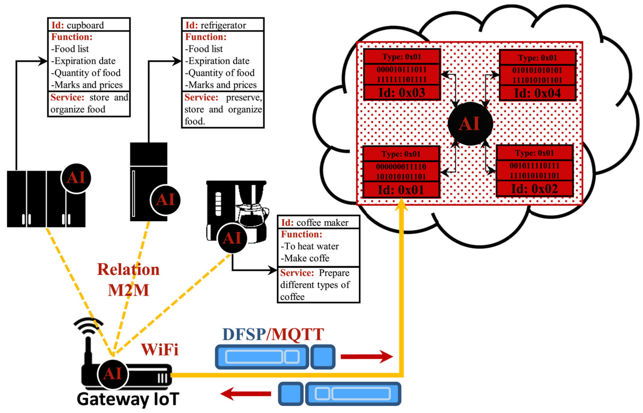


FIGURE 7. Operation of the proposal.

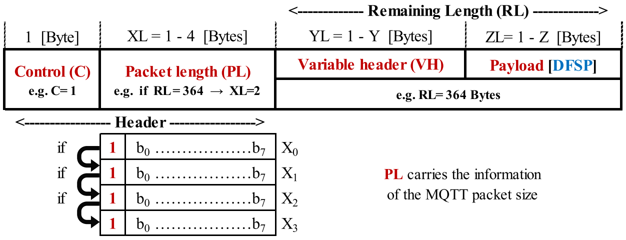


FIGURE 8. MQTT packet format.

TABLE 1. Equations of the algorithm RL.

Message MQTT	Remainning Length (RL), Bytes
CONNECT	$RL = 2 + (\sum_{i=1}^{n} ProtocolVersion[n]) + 1 + 1 + 2 + 2 + (\sum_{i=1}^{n} ClientID[n])$
CONNACK	$RL = 2$
PUBLISH	$RL = 2 + (\sum_{i=1}^{n} TopicName[n]+Payload[n])$
PUBACK	$RL = 2$
SUBSCRIBE	$RL = 2 + (\sum_{i=1}^{n} TopicName[n]) + 1$
SUBACK	$RL = 2$
UNSUBSCRIBE	$RL = 2 + (\sum_{i=1}^{n} TopicName[n])$
UNSUBACK	$RL = 2$
DISCONNECT	$RL = 0$

Cisco Packet Tracer simulator was the best option in this case, although it has limitations when emulating the speed of processing of the packets and the time of decision in the AI algorithm. This simulator allows illustrating the behavior of the RPi3 in a network environment and seeing the operation of the MQTT protocol. It also allows you to modify the pre-installed sample programs and adjust them to the design of this proposal.

The modified program was the MQTT Client and MQTT Broker protocol in Python code into each RPi3. An algorithm was added to both to calculate the length of each of the 16 messages of the MQTTv3.1.1 protocol [30], called Remaining Length (RL). This algorithm can also be understood mathematically knowing that the MQTT packet or message format consists of a Header (always present) + Variable Header (not always present) + Payload (not always present) [31]. The above depends on the type of message. The following figure illustrates the MQTT packet format.

The MQTT packet size (PS) is calculated according to the type of message through the following equation (1).

$$PS = [C + XL] + RL, \text{Bytes} \tag{1}$$

The length of the Control field (C) will always be 1 Byte and if the value to be saved in the Packet length field (PL) is less than or equal to 127, then the header will have a fixed value of 2 Bytes. However, if the value calculated by the RL is greater than 127, the length of the PL field changes, modifying the total size of the package (PS).

The packet length field (PL) saves the size of the calculated packet and can occupy a length of between 1 and 4 bytes. To know how many bytes it is used and how is to save the size in this field, it used the following equation (2).

$$XL = \sum_{n=0}^{3} X_n {}^* 128^n \tag{2}$$

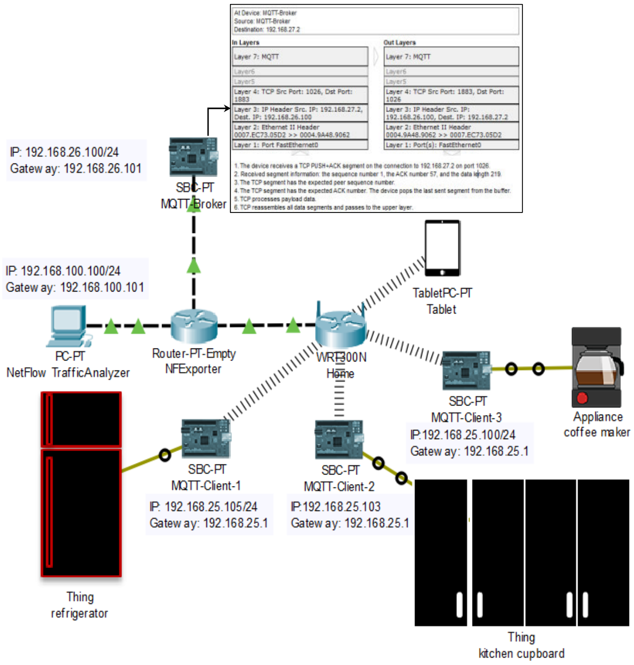
XL is the calculation in Bytes needed to represent the number that corresponds to the value of RL, and that will be transported within the PL field located in the header. The representation of this number in Hex can occupy a space of the range between 1-Bytes to 4-Bytes. Since the length of this field is 8-bit linear, Xn is used to evaluate with the restrictions of the equation (3), if this number requires more than 8 bits to be represented. For this, the 7 least significant bits (LSB) are used and bit 8 is left as carry to indicate the next field as shown in Figure 8.

$$f : \mathbb{R} \rightarrow \{1, 2, 3, 4\} = \#\text{Bytes}$$
$$XL = \begin{cases} 1, & \text{if } 0 < X_n \leq 127 \\ 2, & \text{if } 128 < X_n \leq 16383 \\ 3, & \text{if } 16384 < X_n \leq 2097151 \\ 4, & \text{if } 2097152 < X_n \leq 268435455 \end{cases} \tag{3}$$

The Remaining Length algorithm (RL) calculates the package size depending on the type of message, in some cases; the variable-header is not present. The equations for each of the cases is shown in table 1.

It was necessary to modify the sample code of the simulator because it did not calculate the length of the TCP packets or the MQTT messages. In this way, MQTT messages carry information about their size in the field called package length. Information that facilitated the capture of the data to then model them graphically. Each equation of table 1 was programmed in python within each MQTT message.

The minimum packet size (MIN-PS) is just 2-Bytes and the maximum packet size (MAX-PS) is 256-MB. E.g. the DISCONNECT message is 2 Bytes. Like the existing condition in the MQTT Broker, if the Client ID contains more than 23 characters, the broker responds to the CONNECT message with a CONNACK return code 2: Identifier Rejected [30]. A condition was created to control the minimum and the maximum message size. Considering that XL = 4-Bytes for any PL between 2-MBytes at 256-MBytes and the MAX-PS is 256-MBytes. Then the payload of a PUBLISH message that corresponding to DFSP is 250-MBytes. PL will carry the information obtained from the calculation of PS = 1M-Byte



At Device: MQTT-Broker
Source: MQTT-Broker
Destination: 192.168.27.2

In Layers	Out Layers
Layer 7: MQTT	Layer 7: MQTT
Layer6	Layer6
Layer5	Layer5
Layer 4: TCP Src Port: 1026, Dst Port: 1883	Layer 4: TCP Src Port: 1883, Dst Port: 1026
Layer 3: IP Header Src. IP: 192.168.27.2, Dest. IP: 192.168.26.100	Layer 3: IP Header Src. IP: 192.168.26.100, Dest. IP: 192.168.27.2
Layer 2: Ethernet II Header 0007.EC73.05D2 >> 0004.9A48.9062	Layer 2: Ethernet II Header 0004.9A48.9062 >> 0007.EC73.05D2
Layer 1: Port FastEthernet0	Layer 1: Port(s): FastEthernet0

1. The device receives a TCP PUSH+ACK segment on the connection to 192.168.27.2 on port 1026.
2. Received segment information: the sequence number 1, the ACK number 57, and the data length 219.
3. The TCP segment has the expected peer sequence number.
4. The TCP segment has the expected ACK number. The device pops the last sent segment from the buffer.
5. TCP processes payload data.
6. TCP reassembles all data segments and passes to the upper layer.

FIGURE 9. Simulation of the IoT Network.

+ 4-Bytes + 250-MBytes without exceeding the MAX-PS. Therefore, the calculation of RL does not affect the DFSP protocol because it already takes into account in PS.

The MQTT Client and MQTT Broker initially worked manually, but now due to this modification, it is the control algorithm in gateway whose controls them automatically. In the payload of the PUBLISH message the algorithm sends the protocol DFSP announcing and sharing information about the services and functions of the Things. So a simple network design was proposed, starting with a group of Things where a specific task is assumed. The network of this test supposes an IoT Gateway based on a centralized management architecture following a star topology.

Figure 9 shows the simulation performed with the Cisco Packet Tracer simulator version 7.2.1.

The IoT Gateway of the simulation is represented as the next set of elements, a wireless router WiFi, a Cisco 2911 router (NFExplorer) and SBC-RPi3 (inside is content the MQTT broker and AI control algorithm). A real implementation would be done only with an SBC-RPi3, who would be integrated by MQTT Broker and the AI control algorithm.

Smart things are also integrated by an SBC-RPi3 and host the MQTT client and the AI algorithm, which is responsible for announcing its functions and services. To simulate the operation of each Smart Things, a component called "Thing" was programmed and connected to an SBC-RPi3. Then in this way, it is possible to simulate the set (Thing + SBC-RPi3) as a refrigerator, a cupboard and a coffee machine.

All these devices are connected with WiFi technology following a star topology that supports the transport of IoT protocols [32]. In this case, the MQTT protocol selected and modified for all the reasons explained above.

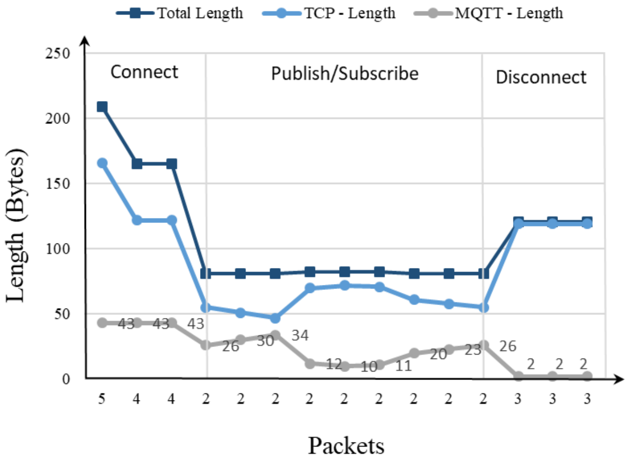


FIGURE 10. Length of the MQTT/TCP packets.

With the target of evaluating the system, it was used to capture the packets in the network, a special configuration with the Cisco 2911 router and a personal computer (PC) with NetFlow Traffic Analyzer [33].

The DFSP protocol is transported over MQTT, which in turn is transported throughout the network over TCP sessions, established at the beginning of the connection. For this reason, all TCP packets are analyzed, as their traffic and bandwidth changes depending on the other protocols.

The capture of the data was achieved with NetFlow Traffic Analyzer through the PC and then was crossing it with the information obtained by the router NFExplorer and the information provided by equation 1 and algorithm RL.

Once all the elements of the network have been configured, is measures the system with two types of tests. Both tests have the same initialization process autonomously through the AI. This test consists in seeing the behavior of the network with or without DFSP protocol [34].

The captured data are analyzed and graphed to interpret it through four graphics.

The three first is composed of three operation mechanisms, connection with the broker (CONNECT message), PUBLICATION/SUBSCRIPTION message and DISCONNECT message. The fourth describes these mechanisms separately in five operation segments.

In Figures, 10 and 11 are observed the behavior in the length of the packets when the network is started without to carry the DFSP protocol and when it is started carrying the DFSP protocol, in both cases, it is realized the same process.

In both figures, the lines that describe the length of the packets corresponding to the connection and disconnection mechanism remain the same.

However, in the mechanism corresponding to subscription and publication the length changes in relation to the data of the DFSP protocol.

The data sent in the DFSP tests were small, only functions corresponding to "on and off", and short services such as "Make Coffee". Therefore, package sizes are not of the order of Mega Bytes, but only of some Bytes.

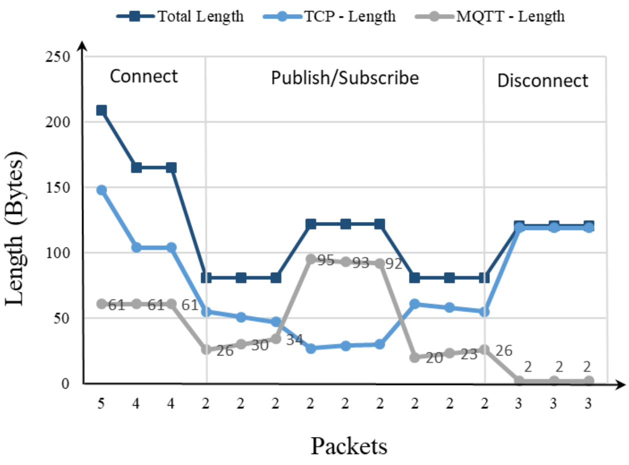


FIGURE 11. Length of the DFSP/MQTT/TCP packets.

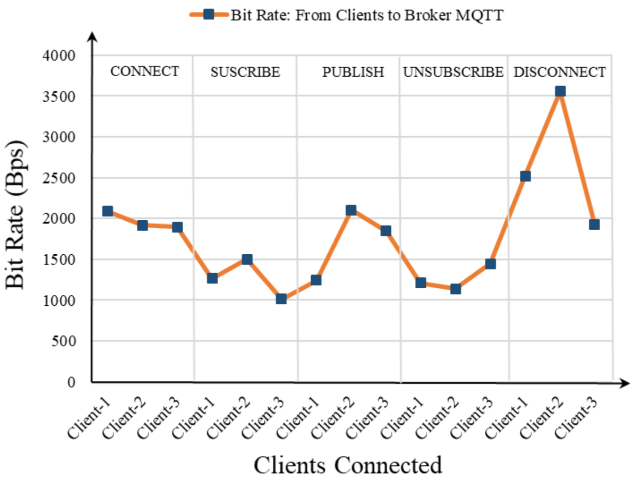


FIGURE 13. Data transmission rate.

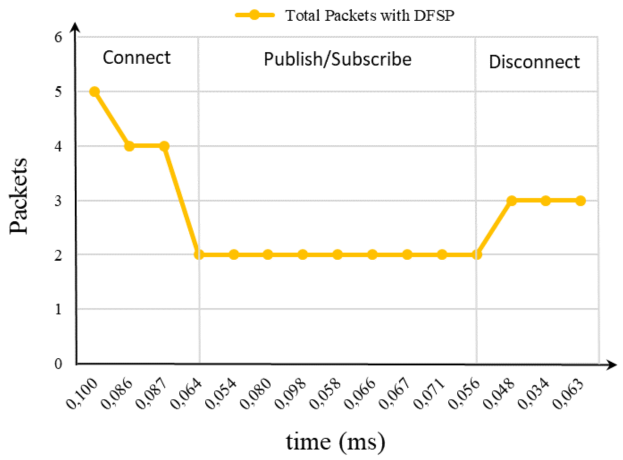


FIGURE 12. Number of packets transmitted.

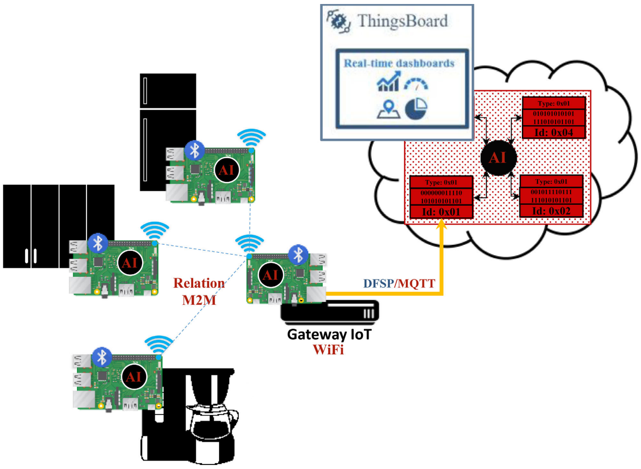


FIGURE 14. Implementation of the proposal.

As can be seen in Figure 11, the length of the packets shows that the DFSP protocol has been introduced in the payload of MQTT compared to what was seen in Figure 10.

The length of the TCP packets given during the test is shown in Figure 11. The maximum values are peaks of 209 bytes, while the minimum value has been 81 bytes, which belongs to an UNSUBSCRIBE message. The average number of bytes transferred has been 116 Bytes.

Figure 12 shows the relation between numbers of packets is transmitted per seconds along the tests. The captured time is the real time of duration each packet sent in the network, which is measured in milliseconds (ms). The maximum time transmission of packets was at 0.1 ms, with 5 packets. The rest of the packets remain constant after the connection is established through the TCP socket.

In this time interval, PUBLISH/SUBSCRIBE messages are exchanged. The average time of the TCP packets transporting them is 0.070 ms.

In Figure 13, it is can see the data transfer rate (DTR) in Bytes per seconds between each Smart Thing and the IoT Gateway. In the simulation, the AI makes a boot of the smart things automatically and start to send the messages of MQTT protocol to the Gateway.

In Figure 13, the transmission of a message different from the MQTT protocol is observed in a sequential test process performed by the AI. This sequence begins with the CONNECT message and ends with the DISCONNECT message. The bit rate of each transmission from the customers is on average between 1000 Bps and 2100 Bps. In the disconnection segment, a higher speed is observed, on average, between 2100 Bps and 3600 Bps.

In the simulation seen in Figure 14 has added to each Thing (fridge, cupboard and coffee maker) a training matrix based on statistics of usage and user preferences.

These data are assumed through a text file, in order to observe the operation of the system and thus be able to capture the packages evidencing the transport of information autonomously to other machines.

The simulation also allows us to observe the functioning of Things by controlling other things with basic functions such as on/off (on, red color), selection of type coffee preparation and the ingredients available for its preparation.

In a real implementation within a smart home, everything will be smart [35], [36]. Each of the things in this proposal, including the gateway, is built with an RPi3 and programmed with AI. As shown in Figure 12, it also connects to the cloud where there is also a platform with AI. As in the simulation, this proposal is expected to use other types of connections such as Bluetooth 5.0, WiFi and Ethernet. The idea is that protocols M2M, in this case, MQTT can be transported throughout the network, regardless of the type of the underlying connection.

V. CONCLUSION

A new algorithm has proposed for allocating and sharing the common resources among the heterogeneous devices in the scenarios of IoT.

Although, the MQTT protocol is redesigned to consume a low bandwidth of QoS. It was demonstrated through simulation that with a simple protocol like DFSP, the length of the payload of the PUBLISH message was sufficient to transport the information about the functions and services of the Things.

Notwithstanding each training matrix of each thing requires many iterations of the user, the simple matrix used that proved that the algorithm AI works ideally in the system.

While there is still no simulator that completely emulates an SBC like the RPi3, the Packet Tracer simulator can be a good option to see the behavior of the algorithms designed within the network.

After successfully modifying the MQTT protocol of the python code example that is in the Packet Tracer simulator, it is likely that the code for the CoAP and HTTP-Rest protocols can also be created and simulated in the network.

Smart Things need to exchange much more information than a sensor node, while there is not a standard protocol that carries more information, as showed in the simulation, working with MQTT is better to achieve the goal.

In future work, we plan verifying that the DFSP algorithm over the MQTT protocol by implementing over using real devices and obtain measurements at runtime and obverse the energy consumption.

The following work consists in showing the results of this system implemented in a real environment, with centralized administration through a Multiprotocol IoT Gateway controlled by AI.

It is expected that in the next results on the real network, it will be possible to observe and measure the designation of M2M relations by the AI and how the AI manages Internet accesses to solve problems.

ACKNOWLEDGMENT

A part of the study has done in the computer science department at the university of sulaimani. At the Univeridad Central (Colombia), to the Smart City work team of the MAXWELL research group, for their special interest in generating new contributions to the networks (IoT & IoE).

REFERENCES

[1] L. García-García, J. M. Jimenez, M. T. A. Abdullah, and J. Lloret, "Wireless technologies for IoT in smart cities," *Netw. Protocols Algorithms*, vol. 10, no. 1, pp. 23–64, 2018, doi: 10.5296/npa.v10i1.12798.

[2] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A survey on enabling technologies, protocols, and applications," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 2347–2376, 4th Quart., 2015, doi: 10.1109/COMST.2015.2444095.

[3] B. Kang, D. Kim, and H. Choo, "Internet of everything: A large-scale autonomic IoT gateway," *IEEE Trans. Multi-Scale Comput. Syst.*, vol. 3, no. 3, pp. 206–214, Jul./Sep. 2017, doi: 10.1109/TMSCS.2017.2705683.

[4] I. Zolotová, M. Bundzel, and T. Lojka, "Industry IoT gateway for cloud connectivity," in *Proc. IFIP Int. Conf. Adv. Prod. Manage. Syst.* Cham, Switzerland: Springer, 2015, pp. 59–66, doi: 10.1007/978-3-319-22759-7_7.

[5] A. Papageorgiou, M. Zahn, and E. Kovacs, "Auto-configuration system and algorithms for big data-enabled internet-of-things platforms," in *Proc. IEEE Int. Congr. Big Data*, Jun./Jul. 2014, pp. 490–497, doi: 10.1109/BigData.Congress.2014.78.

[6] B. Kang and H. Choo, "An experimental study of a reliable IoT gateway," *ICT Express*, vol. 4, no. 3, pp. 130–133, 2018.

[7] C.-L. Zhong, Z. Zhu, and R.-G. Huang, "Study on the IoT architecture and gateway technology," in *Proc. 14th Int. Symp. Distrib. Comput. Appl. Bus. Eng. Sci. (DCABES)*, Aug. 2015, pp. 196–199.

[8] S. Kesavan and G. K. Kalambettu, "IoT enabled comprehensive, plug and play gateway framework for smart health," in *Proc. 2nd Int. Conf. Adv. Electron., Comput. Commun. (ICAECC)*, 2018, pp. 1–5.

[9] V. Kulik and R. Kirichek, "The heterogeneous gateways in the industrial Internet of Things," in *Proc. 10th Int. Congr. Ultra Mod. Telecommun. Control Syst. Workshops (ICUMT)*, 2018, pp. 1–5.

[10] J. Santos, J. J. P. C. Rodrigues, B. M. C. Silva, J. Casal, K. Saleem, and V. Denisov, "An IoT-based mobile gateway for intelligent personal assistants on mobile health environments," *J. Netw. Comput. Appl.*, vol. 71, pp. 194–204, Aug. 2016.

[11] A. P. Athreya, B. DeBruhl, and P. Tague, "Designing for self-configuration and self-adaptation in the Internet of Things," in *Proc. 9th IEEE Int. Conf. Collaborative Comput., Netw., Appl. Worksharing*, Oct. 2013, pp. 585–592.

[12] K. Rajaram and G. Susanth, "Emulation of IoT gateway for connecting sensor nodes in heterogenous networks," in *Proc. Int. Conf. Comput., Commun. Signal Process. (ICCCSP)*, 2017, pp. 1–5.

[13] V. Cerf and R. Kahn, "A protocol for packet network intercommunication," *IEEE Trans. Commun.*, vol. COM-22, no. 5, pp. 637–648, May 1974, doi: 10.1109/TCOM.1974.1092259.

[14] A. E. Khaled and S. Helal, "Interoperable communication framework for bridging RESTful and topic-based communication in IoT," *Future Gener. Comput. Syst.*, vol. 92, pp. 628–643, Mar. 2019, doi: 10.1016/j.future.2017.12.042.

[15] P. Thota and Y. Kim, "Implementation and comparison of M2M protocols for Internet of Things," in *Proc. 4th Int. Conf. Appl. Comput. Inf. Technol./3rd Int. Conf. Comput. Sci./Intell. Appl. Inform./1st Int. Conf. Big Data, Cloud Comput., Data Sci. Eng. (ACIT-CSII-BCD)*, Las Vegas, NV, USA, Dec. 2016, pp. 43–48, doi: 10.1109/ACIT-CSII-BCD.2016.021.

[16] U. Hunkeler, H. L. Truong, and A. Stanford-Clark, "MQTT-S—A publish/subscribe protocol for wireless sensor networks," in *Proc. 3rd Int. Conf. Commun. Syst. Softw. Middleware Workshops (COMSWARE)*, 2008, pp. 791–798, doi: 10.1109/COMSWA.2008.4554519.

[17] P. L. G. Ramirez, J. Lloret, T. Miran, and J. Tomás, "Architecture to integrate iot networks using artificial intelligence in the cloud," in *Proc. 5th Annu. Int. Conf. Comput. Sci. Comput. Intell. (CSCI)*. Las Vegas, NV, USA: IEEE Computer Society, Dec. 2018, pp. 996–1001. doi: 10.1109/CSCI.2018.00192.

[18] J. Lloret, S. Sendra, P. L. González, and L. Parra, "An IoT group-based protocol for smart city interconnection," in *Ibero-American Congress on Information Management and Big Data* (Communications in Computer and Information Science), vol. 978, S. Nesmachnow and C. L. Hernández, Eds. Cham, Switzerland: Springer, Sep. 2019, pp. 164–178, doi: 10.1007/978-3-030-12804-3_13.

[19] R. Pi. (2019). *Raspberry Pi 3 Model B+: Small Single-Board Computers (SBC)*. [Online]. Available: https://www.raspberrypi.org/products/

[20] (2019). *ThingsBoard IoT Platform*. [Online]. Available: https://thingsboard.io/

[21] Mathworks. (2019). *ThingSpeak Internet of Things*. [Online]. Available: https://thingspeak.com/

[22] Google. (2015). *Android Things*. [Online]. Available: https://developer.android.com/things/

[23] Google. (2019). *Android Studio*. [Online]. Available: https://developer.android.com/studio/

[24] Thingsquare. (2016). *Get Started With Contiki and Cooja*. [Online]. Available: http://www.contiki-os.org/start.html

[25] T. Issariyakul and E. Hossain, "Introduction to network simulator 2 (NS2)," in *Introduction to Network Simulator NS2*. Boston, MA, USA: Springer, 2009, pp. 1–18, doi: 10.1007/978-1-4614-1406-3.

[26] J. Grossmann, D. Ziajka, and P. Pękala. (2019). *GNS3 Simulator*. [Online]. Available: https://www.gns3.com/

[27] Cisco Academy. (2019). *Packet Tracer Simulator | Networking Academy*. [Online]. Available: https://www.netacad.com/courses/packet-tracer

[28] Proteus. *Raspberry Pi | Visual IoT Programming-Proteus Design Suite*. Accessed: Jan. 2019. [Online]. Available: https://www.labcenter.com/raspberry_pi/

[29] Microsoft Azure. *Raspberry Pi Azure IoT Web Simulator*. Accessed: Jan. 2019. [Online]. Available: https://azure-samples.github.io/raspberry-pi-web-simulator/

[30] IBM and Eurotech. (2010). *MQTT Version 3.1 Protocol Specification*. [Online]. Available: https:// https://public.dhe.ibm.com/software/dw/webservices/ws-mqtt/mqtt-v3r1.html

[31] A. Steve. (2018). *Understanding the MQTT Protocol Packet Structure*. Accessed: Jan. 2019. [Online]. Available: http://www.steves-internet-guide.com/mqtt-protocol-messages-overview/

[32] R. P. Egli. (2015). *An Introduction to MQTT, a Protocol for M2M and IoT Applications*. [Online]. Available: https://Indigoo.com

[33] Paessler. (2019). *NetFlow Traffic Analyzer*. [Online]. Available: https://www.paessler.com/netflow_monitoring

[34] K. Urunov, J.-I. Namgung, S. Y. Shin, and S.-H. Park, "The interworking functions based on service discovery protocols (SDP) for constrained environment M2M/UIoT communication," in *Proc. IEICE Conf.*, 2016, pp. 742–745.

[35] B. Boyd, J. Gauci, M. P. Robertson, N. Van Duy, R. Gupta, V. Gucer, and V. Kislicins, *Building Real-Time Mobile Solutions With MQTT and IBM MessageSight*, 1st ed. Armonk, NY, USA: International Business Machines Corporation, 2014.

[36] J. Rocher, M. Taha, L. Parra, and J. Lloret, "IoT sensor to detect fraudulent use of dyed fuels in smart cities," in *Proc. 5th Int. Conf. Internet Things, Syst., Manage. Secur.*, 2018, pp. 86–92.

PEDRO LUIS GONZÁLEZ RAMÍREZ received the M.Sc.-Ing. degree. He is currently pursuing the M.Sc. degree with Pontificia Universidad Javeriana, Colombia, and the Ph.D. degree in telecommunication with the Polytechnic University of Valencia, Spain. He is a Lecturer with the Department of Electronic Engineer, Universidad Central, Colombia. He has special interest in android application development and research in telecommunications network engineering, the IoT and IoE, network architecture, protocols, and algorithms. He is also a member of the Research Group Maxwell of Universidad Central and a Junior Researcher in Colciencias, Colombia. He revised many articles as a reviewer and session's chairs in many international conferences.

MIRAN TAHA (Member, IEEE) received the M.Sc. degree from the University of Sulaimani and the Ph.D. degree in telecommunication from the Polytechnic University of Valencia. He served as a Decider of the Computer Department, UNIVSUL, in 2011. He is currently a Lecturer with the Department of Computer Science, University of Sulaimani. He does research in telecommunications network engineering, the IoT, adaptive multimedia streaming, bigdata, and algorithms. He published multiple articles in international conferences and international journals. He has been a member of IEEE Comsoc, ACM, and European Alliance for Innovation (EAI), since 2016. He is also a member of the Integrated Management Coastal Research Institute, Spain. He revised many articles as a reviewer and session's chairs in many international conferences.

JAIME LLORET (Senior Member, IEEE) received the B.Sc. and M.Sc. degrees in physics, in 1997, the B.Sc. and M.Sc. degrees in electronic engineering, in 2003, and the Ph.D. degree (Dr.-Ing.) in telecommunication engineering, in 2006. He worked as a Network Designer and an Administrator in several enterprises. He is currently an Associate Professor with the Polytechnic University of Valencia. He is a Cisco Certified Network Professional Instructor. He is the Chair of the Integrated Management Coastal Research Institute (IGIC), and he is the Head of the Active and Collaborative Techniques and Use of Technologic Resources in the Education (EITACURTE) Innovation Group. He was the Director of the University Master "Digital Post Production," from 2012 to 2016. He is the Director of the University Diploma "Redes y Comunicaciones de Ordenadores." He has authored 22 book chapters and has more than 480 research articles published in national and international conferences, and international journals (more than 220 with ISI Thomson JCR). He is an ACM Senior Member and IARIA Fellow. He was the Vice-Chair for the Europe/Africa Region of Cognitive Networks Technical Committee (IEEE Communications Society), from 2010 to 2012, and the Vice-Chair of the Internet Technical Committee (IEEE Communications Society and Internet society), from 2011 to 2013. He was the Internet Technical Committee Chair of the IEEE Communications Society and Internet society, from 2013 to 2015. He has been the General Chair (or Co-Chair) of 52 International workshops and conferences. He is currently the Chair of the Working Group of the Standard IEEE 1907.1. Since 2016, he has been the Spanish Researcher with highest H-index in *Telecommunications* journal list according to Clarivate Analytics Ranking. He has been involved in more than 450 program committees of international conferences, and more than 150 organization and steering committees. He has led many local, regional, national, and European projects. He is the Editor-In-Chief of *Ad Hoc and Sensor Wireless Networks* (with ISI Thomson Impact Factor), the international journal *Networks Protocols and Algorithms*, and *International Journal of Multimedia Communications*. Moreover, he is an Associate Editor-in-Chief of *Sensors* in the Section Sensor Networks, he is an advisory board member of *International Journal of Distributed Sensor Networks* (both with ISI Thomson Impact factor), and he is the *IARIA Journals* Board Chair (eight journals). Furthermore, he is (or has been) an Associate Editor of 46 international journals (16 of them with ISI Thomson Impact Factor). He has been the Co-Editor of 40 conference proceedings and a Guest Editor of several international books and journals.

JESÚS TOMÁS was graduated in computer science at the Polytechnic University of Valencia, in 1993, getting the best ratings. He finished his Doctoral Thesis, in 2003. He worked as a Software Programmer in several enterprises and as a freelance. Since 1993, he has been an Associate Professor with the Polytechnic University of Valencia. He is a member of the Integrated Management Coastal Research Institute. He is the Director of University Master Develop of mobile applications. He has published multiple articles in national and international conferences and has multiple articles in international journals (more than 17 of these are included in *Journal Citation Report*). He has been involved in several research projects related to public and private pattern recognition and artificial intelligence applied to multiple subjects (four of them as Principal Investigator). His research interests include statistical translation, artificial intelligence, pattern recognition, and sensors networks.