外文文献译文

使用MQTT通信协议的基于物联网的智能家居试验台

Fikret Yalçınkaya, Hüseyin Aydilek, Mustafa Yasin Erten, Nihat İnanç

**摘要：**技术在世界范围内日益发展。此外，科技的发展也为生活的各个领域带来了创新和便利。然而，确保这些由技术带来的创新的连续性揭示了不同的问题。智能家居方式，增加了人类居住区域的质量，是近年来最受欢迎的工作课题之一。在以物联网技术产品打造的智能家居中，传感器和控制设备以更安全的方式进行通信，以协调的方式工作，以更安全的方式保证生态系统的连续性非常重要。本研究利用物联网通信协议中使用最多的MQTT通信协议实现了一个基于物联网的智能家居试验台。利用所开发的系统，通过移动应用控制智能家居的控制和系统的运行性能。利用所开发的系统，通过移动应用控制智能家居的控制和系统的运行性能。根据测试提供的数据得到的结果表明，采用MQTT通信协议开发的系统能够成功地保证智能家居应用中的数据流和控制。

**关键词：**智能家居，物联网，MQTT，试验台

1. **介绍**

技术每天都在快速发展。随着科技的发展，各种提供创新和便利的技术装置走进了我们的生活。自动化系统，不受人为干预，随着技术的发展，已经开始在人类生活中定期发生。

人类从创造开始就需要庇护所，并开始创造与之相关联的空间。自然环境迫使人类这样做，人类开始建造自己的结构，以显示他们所处的时代的特点及其社会和文化的变化。如今，随着现代技术的主导效应，“智能家居”的新概念开始浮现。以智能家居自动化理念，旨在为住房需求既能提供能源时间效率，又能提供安全、舒适、节约等效益。

智能住宅是上世纪50年代普希巴顿大厦遇到的第一个例子，随着时间的推移，它不断获得各种功能。Push Button Mansion是第一个智能家居应用程序，它有很多功能，如下雨时关闭的窗户、自动打开到按铃机的照明、可以根据行驶路径打开和关闭的车库门、火灾和盗窃报警系统(Railton，1950)等。近年来的研究不仅仅是在智能家居中添加功能，还包括能源与废弃物管理(Han等人，2014年)、(周晓等，2016)、(Anvari-Moghaddam等，2016)、系统安全(Kumar，2014)、(Komninos，2014)、(Jacobsson等人，2016年)、监控(Liu等人，2016年)、(Vanus等人, 2017年)、应用开发(Aminikhanghahi等人，2018)、(冯军等人,2017年)、(朱万等,2019)以及新通信协议在系统上的运行。

智能家居这种自动化可以从单点管理我们的空间，借助房内或房外使用的智能传感器和智能家居助手，获得(添加)各种安全功能。智能家居应用程序中使用的所有设备和传感器都由主控家用自动化控制器控制，一般称为智能家居系统。智能家居系统中的所有设备都应该按照定义相互通信，必要时与主控制器通信。智能设备通过网络相互通信，揭示了物联网的概念。

物联网可以定义为一个网络系统，允许电子设备之间使用各种通信协议进行通信。它是1999年由凯文·阿什顿(Kevin Ashton)在演讲中首次提出的，并随着技术的发展而成为现在的形式。据研究显示，2003年互动式设备数量大概为5亿台，但如今已达到140亿台，预计到2020年，这一数字将增至500亿台。

物联网系统开始在许多主题下工作。这些研究可以归结为物联网建立的系统的安全性、通信协议的性能以及涉及各个领域的应用。(Alrawais等人, 2017年)、(Nurse等人，2017年)、(Dabbagh&Rayes, 2019 )已在物联网安全(Lee等, 2013 )、(Johnsen等人,2013年)、(Luzuriaga等人, 2014年)、(Luzuriaga等人, 2015年)方面开展工作。物联网系统所用通信协议的文献综述及性能比较见表1。

在物联网系统中，设备/物件(发行者/订阅者)之间或与代理之间的通信只能通过使用相同的网络协议来实现。设备之间的相互通信引发了不同的问题。网络协议用于设备的通信；网络流量、传输数据量、传输数据量，以及物联网设备硬件功能有限的问题，在重要硬件问题上存在有效的硬件和电源问题。为此，企业开发了各种网络协议，使物联网设备能够更高效地进行通信。

1. **材料与方法**
   1. **IoT**

近年来，随着技术的发展，电子器件获得了直接相互通信和共享数据的能力，或者通过创建一个生态系统，电子器件之间通过各种网络结构和协议进行通信和共享数据的能力随着时间的推移被命名为物联网。

各种具有连接能力的智能电子设备可以成为物联网生态系统的一部分。随着这些设备数量的不断增加，我们在智能家居、智慧环境、智慧城市、智慧卫生服务、智慧农牧业等多个领域遇到了应用，最后是智慧工业和军事应用。

在一个硬件特性受限的物联网生态系统中，由于通信流量增加、功耗增加、处理器、存储器和存储容量不足等原因，通信协议变得尤为重要。为此，出现了MQTT、CoAP、AMQP、SOAP等网络协议。

* 1. **MQTT**

消息队列遥测传输(MQTT)是IBM于1999年提出的一种基于消息的通信协议。它是物联网平台内机器之间交换数据最常用的协议。这种协议之所以首选，是因为通信数据包的大小较小，易于使用和实现，网络带宽较大。

MQTT是一种基于发布/订阅系统的协议。该系统由客户端、经纪人和主题组成。客户端表示连接到网络的设备以及发送和接收数据，如传感器、手机和计算机。MQTT客户端既可以是发行者，也可以是订阅者。发布者是系统上要发送的数据的来源。订阅者是希望接收发布者消息的客户端。MQTT代理是连接系统中客户端的设备。当与其相连的客户端发布数据并进行适当处理时，经纪人负责收集和维护该数据。然后将此数据传送给与其相连的订阅者。广播者和订阅者不需要同时工作和了解对方。代理在客户端离线时将数据按顺序检索到缓冲区中，在客户端在线时转发。它有这样的排队系统。

MQTT有一个三级控制包来建立安全的通信：强制固定头、可选变量头和负载。可选字段常常使协议的操作复杂化。CONNECT、CONNACK、PUBLISH、PUBACK、PUBREC、PUBREL、SUBSCRIBE、SUBACK和MQTT是MQTT客户端和MQTT代理之间交换的一些MQTT控制包。

在MQTT客户端和MQTT代理之间建立了成功的网络后，客户端和代理之间交换了控制包。希望连接到MQTT代理的客户端发送一个CONNECT包，指定其身份、标志、协议级别和其他字段。服务器通过CONNACK包确认客户端，其返回代码表示连接状态。这确保了客户端和代理之间的连接。

如果客户端想以发布者的身份发送数据，它将向代理服务器发送一个PUBLISH包。该包包括QoS传输级别、主题名、负载容量等。如果要发送的数据在QoS0中传输，则经纪人不会为广播包发送任何确认包。如果想要将数据传送到QoS1，则经纪人向客户端确认数据包已经与PUBACK包一起广播。在QoS2中，4个包被交换。经纪人确认PUBLISH包是与PUBREC包一起收到的。然后MQTT客户机发送它想要用PUBREL包发布的数据请求。经纪人随后发送PUBCOMP的第四个数据包，确认消息完整，并在给定主题上发布一次，表明事务完整。

如果MQTT客户端想要接收已发布的消息，它将发送SUBSCRIBE包给代理。经纪人用SUBACK包确认对请求的订阅。当订阅成功时，指定主题上的消息将以QoS传输级别传输给订阅者。为了解订一个主题，订阅者向经纪人发送UNSUBSCRIBE包，经纪人用UNSUBACK包确认取消订阅。经过一定的超时传输，客户与经纪人之间的连接终止。客户端将PINGREQ包发送给服务器以维护连接，表示它是活动的。MQTT代理用PINGRESP包响应订阅者，并确保连接继续。

MQTT客户端向代理发送DISCONNECT包以终止连接。代理并没有针对这个数据包向订阅者发送应答包，但客户端随后删除所有与订阅者相关的消息，订阅者与代理断开。

MQTT使用QoS级别来传输消息。QoS是一种基于消息关于数据分布保证的双边协议的协议。虽然TCP / IP提供了保证的数据呈现方式，但如果一个TCP连接断了，传输的消息丢失，可能会发生数据丢失。因此，MQTT在TCP的顶层增加了3个QoS级别：

a ) QoS0 (最多一次)：在这个QoS级别上，消息被发送一次，并且消息的发送没有保证。

b ) QoS1 (至少一次)：在这个QoS级别上，数据被发送一次。通过设置消息的重复次数，可以不止一次地发送消息。

c ) QoS2 (确切地说是一次)：在这个QoS级别上，使用4路握手系统，消息精确地一次发送。

1. **开发的应用程序**

本研究实现了一个基于物联网系统架构的智能家居系统试验台，根据MQTT协议保证物/设备之间的通信。系统的结构如图3所示。

开发的智能家居系统由传感器和控制元件(订阅服务器/发布服务器)和主中央处理单元(Broker)组成。在开发的系统中，PIR传感器用于运动检测，MQ4传感器用于气体泄漏检测，DHT11传感器用于测量热湿，IR火焰传感器用于火灾检测，光刻胶(LDR)用于测量光强，ACS712用于测量电流。继电器、报警、门锁、水阀和气阀作为控制元件。

* 1. **建立代理**

在应用环境中，以ARM微处理器的树莓Pi3单板计算机为主要处理单元。此卡因功耗低、物理尺寸小、使用方便而被选中。在卡上安装了基于Debian的Raspbian操作系统作为操作系统。Apache Web服务器用于Web服务和数据库。遵循操作系统的一般配置，使用Eclipse Foundation开发的开源Mosquitto MQTT代理来管理MQTT消息内容。Mosquitto由于其开源结构而受到青睐，它允许进行各种修改和添加以及丰富的文档。MQTT服务器和Apache Web Server集成是通过开发的程序提供的，以保证来自物联网环境的每一个数据都以时间标签、消息标签和内容保存在数据库中。此外，开发的程序对物联网环境中采集到的数据进行过滤，并发送确定在QoS2级别具有重要意义的数据。

* 1. **建立发布者**

通过运动传感器、气体传感器、火焰传感器、湿度传感器、温度传感器、LDR和电流传感器从物联网环境中采集的数据，这些被称为发布者元素，借助NodeMCU微控制器读取并转换为有意义的数据，发送给代理。图4给出了发行者元素的一般工作流程图。

* 1. 建立订阅者

开发的系统采用继电器、报警、门锁、水阀和气阀作为订阅元件。利用Node MCU微控制器对物联网生态系统采集到的数据进行分析，并将适当的命令传递给相关的控制元素。订户元素的一般工作流程图如图5所示。

* 1. 移动应用开发

如今的技术基础设施使得远程访问和控制许多设备变得容易。对于一个智能家居系统来说，远程监控是一个重要的需求。开发了一个移动应用程序，可远程监控测试环境中的数据，并对控制元件进行远程控制。开发的移动应用程序是为双向操作而设计的。如果移动设备连接到与智能家居相同的Wi - Fi网络上，那么移动应用充当物联网生态系统的一部分，同时充当发布者和订阅者。如果移动设备通过不同的网络接入互联网，则设备可以通过Web服务对生态系统进行控制，并对数据进行检查。该移动应用是按照Android移动设备开发的。应用程序的截图如图6所示。

1. **结果**

测试环境被测试并在各种场景下运行。所创建的场景概述如下。

场景1：单发布者和单订阅者是活动的

在此场景中，发行者和订阅者首先为自己的案例检查系统工作和性能。然后在发行者和订阅者同时工作的情况下对系统进行了测试。除此之外，移动设备首先通过连接到同一Wi-Fi网络并连接到不同的网络进行单独测试。该移动设备通过了测试，没有任何问题，即使在出版商和用户的情况下也通过了与系统的连接测试。在只有一个发布者和一个订阅者的情况下，系统工作没有任何问题。

场景2：两个发布者和两个订阅者都是活动的

在这个场景中，系统被测试为两个发布者和两个参与者积极工作。在此场景中，移动应用程序通过连接同一Wi-Fi网络和不同网络进行测试。在有两个发布者和两个参与者的情况下，系统工作没有任何问题。

场景3：所有发布者和所有订阅者都是活动的

在此场景中，测试系统中所有（7）发布者和所有（5）参与者的状态。除此之外，移动应用程序还通过连接同一Wi-Fi网络和通过不同Wi-Fi网络集成到系统中。虽然移动应用程序连接在同一个Wi-Fi网络上，但在通信和控制方面没有问题。然而，当移动应用接入不同的Wi-Fi网络时，观察到应用监控功能不存在问题，而控制功能存在延迟。

1. **结论**

本研究利用MQTT通信协议实现了一个基于物联网技术的智能家居测试环境。在物联网生态系统中添加了各种传感器和控制元件，测试了MQTT通信协议在不同场景下的性能。所开发的机制在所有的发布者和参与者相互连接的场景中都得到了测试，并且观察到其工作没有任何问题。开发的测试环境在移动应用程序连接到多个订阅者和发布者以外的网络的情况下，在控制函数中遇到了延迟问题。它已在所有其他测试场景中成功工作，并成功通过测试。测试应用结果表明，MQTT通信协议适用于物联网智能家居系统。

人们认为，上述问题可以通过对移动应用程序的改进来克服。在今后的研究中，可以进行MQTT和CoAP通信协议比较和系统缩放的研究。

外文文献原文

IoT based Smart Home Testbed using MQTT Communication Protocol

Fikret Yalçınkaya, Hüseyin Aydilek , Mustafa Yasin Erten, Nihat İnanç

**ABSTRACT：**Technology is developing day by day in the world. In addition, developing technologies bring innovations and conveniences to all areas of life. However, ensuring the continuity of these innovations brought by technology reveals different problems. Smart home approach, which increases the quality of human living areas, is one of the most popular working subjects of recent times. In a smart home, built with the Internet of Things technology products, it is very important that the sensors and control devices communicate in a safer way and work in a coordinated manner to ensure the ecosystem's continuity in a safer way. In this study, an IoT based smart home testbed was realized by using MQTT communication protocol which one of the most used IoT communication protocols. With the developed system, the control of the smart home and the operating performance of the system were controlled with the mobile application. The results obtained in the light of the data provided by the test show that the system developed with the MQTT communication protocol can successfully ensure data flow and control in Smart Home applications.

**Keywords:** Smart Home, IoT, MQTT, Testbed

1. **INTRODUCTION**

Technology is developing rapidly on a daily basis. Together with the developing technology, various technological devices that provide innovation and convenience come into our lives. Automation systems, free of human intervention, have started to take place in human life regularly with the development of technology.

Mankind has needed shelter starting with the creation and has begun to create spaces associated with it. Natural environments forced human to do that and, mankind began to construct their own structures that show the characteristics of the era in which they live and their social and cultural change. Nowadays, the concept of "smart home" has started to emerge with the dominant effect of modern-era technology. With smart home automation concept, it is aimed to provide both energy-time efficiency as well as security, comfort and saving benefits for housing needs.

Smart houses, the first example encountered with the Push Button Mansion in the 1950s, continue to gain various features over time. Push Button Mansion was the first smart home application, it had many features such as windows that close when it rains, lighting that opens automatically to the bell press, garage door that can open and close according to the driving path, fire and burglar alarm systems (Railton, 1950). Studies in recent years are not only about adding features to smart homes, but also energy and waste management (Han et al., 2014), (Zhou et al., 2016), (Anvari-Moghaddam et al., 2016), system security (Kumar, 2014), (Komninos, 2014), (Jacobsson et al., 2016), monitoring (Liu et al, 2016), (Vanus et al., 2017), application development (Aminikhanghahi et al., 2018), (Feng et al., 2017), (Juwan et al., 2019) and the operation of new communication protocols on the system.

Smart home kind of automation can manage our spaces from a single point and gain (add) various security features with the help of smart sensors and smart home assistants used inside or outside the house. All devices and sensors used in the smart home applications are controlled by the master home automation controller, generally called a smart home system. All devices within the smart home system should communicate with each other, by definition, and with the main controller when necessary. The communication of smart devices with each other over a network revealed the concept of Internet of Things (IoT).

IoT can be defined as a network system that allows electronic devices to communicate with each other using various communication protocols. It was first put forward in a presentation by Kevin Ashton in 1999 and it has become the present form with the developing technology. According to the research, the number of devices interacting with each other in 2003 was 500 million but today it has reached to 14 billion and it is estimated that this number will increase to 50 billion by 2020.

IoT systems started to work under many topics. These studies can be classified as the security of a system established with IoT, performances of communication protocols and applications that are included in various fields. (Alrawais et al., 2017), (Nurse et al., 2017), (Dabbagh & Rayes, 2019) have worked on IoT security (Lee et al., 2013), (Johnsen et al, 2013), (Luzuriaga et al ., 2014), (Luzuriaga et al., 2015). The literature review of the communication protocols used in IoT systems and the comparison of their performances is given in Table 1.

In an IoT system, the communication of devices/things (Publisher / subscriber) with each other or with the broker can only be possible by using the same network protocol. The communication of the devices with each other raises different problems. Network protocol used for communication of devices; network traffic, the amount of data transmitted, the transmission of data, as well as problems with the limited hardware features of the IoT devices with effective hardware and power problems in the issue of important hardware. For this reason, various network protocols have been developed by companies to enable IoT devices to communicate more efficiently.

1. **Material and Methods**

**2.1. IoT**

Electronic devices have gained the ability to communicate and share data directly with each other or by creating an ecosystem with the development of technology in recent years, The ability of electronic devices to communicate with each other and to share data using various network structures and protocols has been named as the Internet of Things (IoT) over time.

All kinds of smart electronic devices with connection capacity can be a part of the IoT ecosystem. With the increasing number of these devices, we encounter applications in many fields such as smart home, smart environment, smart city, smart health services, smart agriculture and animal husbandry, and finally smart industrial and military applications.

In an IoT ecosystem built with constrained hardware features, the communication protocol has gained particular importance due to reasons, such as increased communication traffic, power consumption, insufficient processor, memory and storage capacity. For this reason, network protocols such as MQTT, CoAP, AMQP, SOAP have emerged.

**2.2. MQTT**

Message Queue Telemetry Transport (MQTT) is a message-based communication protocol introduced by IBM in 1999. It is the most common protocol used to exchange data between machines within the Internet of Things platform. This protocol is preferred because the size of the communication packets is small, easy to use and implement, and the network bandwidth is large.

MQTT is a protocol based on the publish / subscribe system. This system consists of Client, Broker and Topic. The client represents the devices which are connected to the network and sending and receiving data such as sensors, cell phones, and computers. An MQTT client can be both a publisher or a subscriber. The publisher is the source of the data on the system to be sent. A subscriber is a client who wants to receive messages from the publisher. A MQTT broker is a device that connects clients in the system. The broker is responsible for collecting and maintaining that data when the client connected to it publishes data and processing it properly. It then transmits this data to the subscribers connected to it. The broadcaster and subscriber do not need to work and know each other at the same time. The broker retrieves the data into the buffer in sequence when the client is offline and forwards it when the client is online. It has a queuing system in this way.

The MQTT has a three-stage control package to establish safer communication: a mandatory fixed header, an optional variable header, and a load. Optional fields often complicate the operation of the protocol. CONNECT, CONNACK, PUBLISH, PUBACK, PUBREC, PUBREL, SUBSCRIBE, SUBACK, and MQTT are some of the MQTT control packages that are exchanged between MQTT clients and the MQTT broker.

After a successful network has been established between the MQTT client and the MQTT broker, the control packages are exchanged between the client and the broker. The client that wants to connect to the MQTT broker sends a CONNECT packet specifying its identity, flags, protocol level, and other fields. The server confirms the client through the CONNACK packet with a return code indicating the connection status. This ensures a connection between the client and the broker.

If the client wants to send data as a publisher, it sends a PUBLISH packet to the broker. This package includes QoS transmission level, subject name, load capacity, etc. If the data to be sent is transmitted in QoS 0, the broker does not send any confirmation packet for the broadcast packet. If the data is transmitted to QoS 1, the broker confirms to the client that the packet has been broadcast with the PUBACK packet. In QoS 2, four packages are exchanged. The broker confirms that the PUBLISH package was received with the PUBREC package. The MQTT client then sends a request for the data that it wants to publish with a PUBREL packet. The broker then sends the fourth packet of PUBCOMP, confirming that the message is complete and published once on the given subject, indicating that the transaction is complete.

If the MQTT client wants to receive the published message, it sends the SUBSCRIBE packet to the broker. The broker confirms the subscription to the request with the SUBACK package. When the subscription is successful, messages on the specified subject are transmitted to the subscriber with the QoS transmission level. To unsubscribe a topic, the subscriber sends the UNSUBSCRIBE package to the broker and the broker confirms the cancellation of the subscription with the UNSUBACK package. After a certain timeout period, the connection between the client and the broker is terminated. The client transmits the PINGREQ packet to the server to maintain the connection, indicating that it is active. The MQTT broker responds to the subscriber with the PINGRESP packet and ensures that the connection continues.

The MQTT client sends a DISCONNECT packet to the broker to terminate the connection. The broker does not send a reply packet to the subscriber in response to this packet, but the client then deletes all messages related to the subscriber and the subscriber is disconnected from the broker.

The MQTT uses QoS levels to transmit messages. QoS is a protocol based on a bilateral agreement of a message with respect to the assurance of data distribution. Although TCP / IP provides guaranteed data presentation, data loss may occur if a TCP connection is broken and transmitted messages are lost. Therefore, MQTT adds 3 QoS levels to the top of TCP:

a) QoS0 (Up to one time): At this QoS level, the message is sent up to one time and delivery of a message is not guaranteed.

b) QoS1 (At least once): At this QoS level, the data is sent at least once. It is possible to send a message more than once by setting the repetition of the message.

c) QoS2 (Exactly once): At this QoS level, the message is sent precisely once using the 4-way handshake system.

1. **Developed Application Testbed**

In this study, a smart home system test-bed based on IoT system architecture, where communication between things/devices is guaranteed according to MQTT protocols, is realized. The architecture of the system is given in Figure 3.

The developed smart home system consists of sensors and control elements (Subscriber / Publisher) and the main central processing unit (Broker). In the developed system, PIR sensor is used for motion detection, MQ4 sensor for gas leak detection, DHT11 sensor for measuring heat and humidity, IR flame sensor for fire detection, photoresistor (LDR) for measuring light intensity and ACS712 for measuring current. Relay, alarm, door lock, water valve and gas valve are used as control elements.

**3.1. Establishment of Broker**

In the application environment, Raspberry Pi 3 single board computer with ARM microprocessor is used as the main processing unit. This card has been chosen due to its low power consumption, physical dimensions and ease of use. Debian based Raspbian operating system was installed as the operating system on the card. Apache web server is installed for Web Service and database. Following the general configuration of the operating system, the open source Mosquitto MQTT broker, developed by the Eclipse Foundation, was used to manage the MQTT message contents. Mosquitto has been preferred because of its open source structure, it allows for various modifications and add-ons as well as rich documentation. MQTT Server and Apache Web Server integration is provided by means of a developed program to ensure that each data from IoT environment is kept in the database with time tag, message tag and content. In addition, the developed program filters the data collected from IoT environment and sends the data that are determined to be significant at the QoS 2 level.

**3.2. Establishment of Publisher**

The data collected from the IoT environment via motion sensor, gas sensor, flame sensor, humidity sensor, temperature sensor, LDR and current sensor, which are called as publisher elements, are read with the help of NodeMCU microcontroller and converted to meaningful data and sent to the broker. The general working flow diagram of the publisher elements is given in Figure 4.

**3.3. Establishment of Subscriber**

Relay, alarm, door lock, water valve and gas valve were used as Subscriber elements in the developed system. The NodeMCU microcontroller was used to analyze the data collected from the IoT ecosystem and to transmit the appropriate command to the relevant control elements. The general working flow diagram of the subscriber elements is given in Figure 5.

**3.4. Development of Mobile Application**

Today's technological infrastructure makes it easy to access and control many devices remotely. For a smart home system, remote monitoring and control is an important requirement. A mobile application has been developed to remotely monitor the data in the test environment and to control the control elements remotely. The developed mobile application is designed for two-way operation. If the mobile device is connected to the same Wi-Fi network as the smart home, the mobile application acts as a part of the IoT ecosystem and acts as both a publisher and a subscriber. If the mobile device accesses to the Internet through a different network, the device can control the ecosystem and examine the data via the web service. The mobile application was developed in accordance with Android mobile devices. The screenshots of the application are given in Figure 6.

1. **Results**

The test environment was tested and run under various scenarios. The scenarios created are summarized below.

Scenario 1: Single Publisher & Single Subscriber are Active

In this scenario, the publisher and the subscriber first checked for the system work and performance for their own case. The system was then tested in cases where both the publisher and the subscriber were working simultaneously. In addition to this scenario, the mobile device was first tested alone by connecting to the same Wi-Fi network and connecting to a different network. The mobile device, which passed the tests without any problems, was tested by connecting to the system even in the case of both the publisher and the subscriber. In cases where there is only one publisher and one subscriber, the system works without any problems.

Scenario 2: Two Publishers & Two Subscribers are Active

In this scenario, the system was tested for two publishers and two participants actively working. The mobile application was tested in this scenario by connecting to both the same Wi-Fi network and different network. In cases where there were two publishers and two participants, the system worked without any problems.

Scenario 3: All Publishers & All Subscribers are Active

In this scenario, the status of all (seven) publishers and all (five) participants in the system is tested. In addition to this scenario, the mobile application is integrated into the system both by connecting to the same Wi-Fi network and over different Wi-Fi networks. While the mobile application was connected to the same Wi-Fi network, there was no problem in communication and control. However, when the mobile application is connected to a different Wi-Fi network, it is observed that there are no problems in the application monitoring functions but there are delays in the control functions.

1. **Discussions**

In this study, an IoT based smart home test environment was realized by using MQTT communication protocol. Various sensors and control elements were added to the IoT ecosystem and the performance of the MQTT communication protocol on different scenarios was tested. The developed mechanism has been tested in all scenarios in which all publishers and participants are connected and it has been observed that it works without any problem. The developed test environment experienced delay problems in control functions in the scenario where the mobile application was connected to a network other than the multiple subscriber and publisher. It has worked successfully in all other test scenarios and passed the tests successfully. As a result of the test application, it was observed that the MQTT communication protocol is suitable for an IoT smart home system.

It is thought that the mentioned problems can be overcome with the improvements to be made on the mobile application. In future studies, it is considered that studies on comparing MQTT and CoAP communication protocols and system scaling can be performed.