QALAAI ZANISTSCIENTIFIC JOURNAL
A Scientific Quarterly Refereed Journal Issued by Lebanese French University – Erbil, Kurdistan, Iraq
Vol. (8), No (5), Winter2023
ISSN 2518-6566 (Online) - ISSN 2518-6558 (Print)

Assessing the Quality of Internet of Things (IoT) Medical Application

Teeba Ismail Kh.
Department of Computer Network, College of Engineering and Computer Science, Lebanese French University, Erbil, Kurdistan Region, Iraq
Email: teeba.ismail@lfu.edu.krd

Banar Fareed Ibrahim
Department of Information Technology, College of Engineering and Computer Science, Lebanese French University, Erbil, Kurdistan Region, Iraq
Email: bnar.fareed@lfu.edu.krd

ARTICLE INFO

Article History:
Received: 26/3/2023
Accepted: 5/7/2023
Published:Winter2023

Keywords: *Internet of things, healthy application, IoT, Model-Based Testing for Performance Attribute, MBTPA.*

Doi:
10.25212/lfu.qzj.8.5.45

ABSTRACT

The number of applications that use internet of things (IoT) concepts is rapidly increasing. Without high-quality assurance, this growth cannot continue. According to the literature, quality assurance (QA) models for internet of things software applications are needed urgently.

The goal of this study is to build a model-based testing framework for assessing the performance quality of one of the IoT applications. A real-world IoT case implementations have been modeled and analyzed in order to determine the output quality attributes. For software quality assurance applications, the International Organization for Standardization (ISO 25000) quality model was used.

The case study consists of thousands of combinations of test cases, which are mostly beyond the capabilities of the resources for achieving the target in the right time. To solve this problem ACTS has been introduced to reduce the case dimensions and find the optimum cases.

Finally, all these techniques have been integrated to a new model-based approach called Model-Based Testing for Performance Attribute (MBTPA). The MBTPA has been applied to the case successfully.

1 Introduction

The internet of things developed gradually and gained importance recently due to its ability to impact daily life and future societies. It is expected that cars, homes, consumer products, industries, and other objects will become managed via an internet connection and data analysis in the near future [1]. This combination of connected objects with data analysis capabilities could be a reliable source for intelligent decisions that could transform lives in the future. The IoT applications extend across a range of security, eHealth, entertainment, smart cities, military, and many other accurate fields [2], and more objects will gain the ability of direct internet connection in the coming years. As this expansion of connected devices and applications develops, the IoT is going to affect our lives and it is a public safety field. Additionally, with its expansion, the chance of system failure in IoT becomes more common. Therefore, it is not acceptable to enhance less quality IoT systems that might cause loss of resources, data or even life. For this reason, Quality Assurance is a significant and worthy issue for the IoT systems before releasing these devices, applications, and systems to market [3].

In the IoT domain, the quality is the degree to which the system satisfies the stated and implied needs of its various users. The requirements of users are performance, security, maintainability, portability, etc. [4]. They are precisely what is represented in the quality model, which classes the product quality into main attributes and sub-attributes.

2 Related Work

During the past decade, significant effort has been spent in researching the field of IoT. One of the most significant areas of research is QA. Recently, several literature studies have been published concerning the aspects related to IoT quality. One of the earlier studies is for Ming and Yan in 2013, which integrated the (IoT) composite service quality specifications, and put forward effective QoS metric decomposition and optimization methods. The authors divided the complex QoS calculation model into four simple models: the serial mode, parallel mode, branch mode, and circulation mode; each model has a computational process. Combined with the QoS technology of the composite service, the authors followed the algorithm to find the sub-optimal service at a reasonable cost under QoS constraints [5]. In the same year, Berhanu et al. tried to use the testbed as a platform for testing the IoT applications. They mentioned setting up a testbed for adaptive security for the project ASSET (Adaptive Security for Smart Internet of Things in eHealth) using existing commercial off-the-shelf products and open-source software [6]. While in 2015, Adjih et al. presented the open testbed FIT IoT-LAB. IoT-LAB is expected to push IoT testing technology growth by providing accurate open-access and open-source multi-user scientific tools.

Foidl and Felderer in 2016, outlined the QA requirements of IoT into six categories: environment, user, service level agreement, organization, security, and data management [9]. Further, they described the QA of IoT applications by subdividing it into four categories: defect prevention, defect analysis, user incorporation, and organizational. In contrast, Banerjee and Sheth in 2017 classified IoT applications quality data into two categories: specifications and conformance. Additionally, they improved the quality of IoT by analyzing two daily life cases requiring a high level of quality [10]. In the same year other authors propose an IoT testbed around city light poles, with modular hardware and software architecture to enable experimentation with different technologies [11]. In the same year, Nair et al. proposed a user-centric and automated IoT testbed in context with a setting that is close to the real-world South African environment to address the specific IoT vulnerabilities in the country. The specific IoT vulnerabilities they focused on in the test are security and reliability [12]. In 2017, Li and Huang used generalized stochastic Petri net (GSPN) modeling techniques to forecast performance assessment for recoverable IoT services. The average response time was a success measure. The approach showed that the average response time will decrease with service rate increases. It also showed that hardware failure is more difficult to manage than software failures and can often be considered disastrous to system stability and performance. This study can be expected to provide an efficient approach to predict IoT services output before implementation. Using GPSN simulation program called PIPE, the experimental results are obtained [13]. In the same year other authors designed a MBT for the IoT communication [14]. Also, they carried out experiments in which there were five models implementations of MQTT brokers/servers, and they examined these models. White et al. produced a systematic mapping of the quality for IoT applications. They focused on the quality of service (QoS) in IoT applications. They provided a view of the state of QoS approaches in IoT by assuming that QoS can be assured at different layers of the IoT architecture. However, they focused on quality aspects from the perspective of QoS and high-level quality standards without going into the detailed methodologies of assuring various quality aspects [15].

In 2018, Wang et al. evaluated the performance of the blockchain-based IoT security model [16]. Simultaneously, Hiun et al. introduced a service-based approach for an automated IoT testing framework regarding coordination, costs, and scalability issues. While in the same year, Ying et al. dealt with the reliability which is one of the software quality attributes. Moreover, they used a continuous-time Markov chain model to evaluate the reliability of Blockchain Based IoT applications [17]. In 2019, Kuroiwa et al. designed a hybrid testing environment between the execution test and model checking for the IoT system [18].

While Nakhkash et al. in same year tried to evaluate the performance of the Wearable Devices and Mobile Gateways in IoT Applications. Nine applications from the LOCUS benchmark have been utilized and tested on different boards having hardware specifications close to wearable devices and mobile gateways. The execution time and energy consumption results of running the benchmark on the boards are calculated mathematically. The results are then used for providing insights for system designers when designing and choosing a suitable computation method for IoT systems to achieve a high quality of service (QoS). The results show that, depending on the application, offloading methods can be used for achieving certain improvements in energy efficiency [19].

Concerning the quality of IoT applications, several mapping and literature studies have been published recently, these studies concentrated on either the big image of IoT quality or individual quality aspects. For example, Ahmed et al. conducted an extensive SMS on most quality aspects covered by the ISO/IEC 25000 standard that is applicable to IoT applications [21]. Lepekhin et al. addressed several quality aspects in their systematic mapping study on IoT challenges [22].

In 2020 Rateb et al. used the MBT framework to test the load and security aspects for an IoT Blockchain-Based vehicles communication [23]. In contrast, Miroslav et al. used an automated model-based approach to test smart TV applications usability and accessibility [24].

2.1 Quality Assurance in IoT Software Applications

Software quality assurance is the process to define how software quality can be achieved and how the manufacture knows that it has reached the standard.

The quality characteristics of the product can be assessed by utilizing the quality model. The design of the quality model requires specifying the quality attributes to assess the features of the software[25] . The quality characteristics are defined based on a set of standards such as ISO/IEC 9126 [26]. These standards are illustrated by the International Organization for Standardization (ISO) to deal with the quality aspects to determine the quality of the software as a quality model [27].

3 The Home Health Hub IoT (H^3IoT)

The H^3IoT is a novelty IoT based layered model that consists of the dependence and interconnection of biosensors, communication networks, microcontroller, gateway, internet, and applications [28] . The design is intended to be useful, efficient, and simple to use. After using it, it becomes easy to track the health status of the older people who live at home by their family, physicians, hospitals, and their doctors [28].

3.1 Home Health Hub IoT Architecture

The H^3IoT architecture is shown in Figure 1. The first layer is the Physiological Sensing Layer (PSL) which is the lower-level layer. The role of PSL is to use sophisticated bio matters to different physiological activities sensitively. Electric cardiovascular (ECG) and electric kernel (EEG) are examples of PSL. (EMG) covers heart, scalp (brain), and muscle electrical activity. Sensors like a panic alarm (cryptic behavior may occur at any time primarily in the event of an accident), weight scale (measurement of weight), pulse oximeter (measurements of blood SpO2 and pulse), blood pressure meter (calculation of blood pressure), blood glycolysis (measures temperature of the body) [28]. There may be a few additional sensors, which if required, may be added later.

The raw sensed data is then sent for additional processing to the upper layers. The Local Communication Layer (LCL) is the second layer, which is of the utmost significance, transferring sensed data to the upper layers at PSL. In essence, PSL communication technology works within a low geo-range (10 to 900 meters). That is because the elderly should benefit from minimal physical interference when interacting with the system as a whole [28].

The Information Processing Layer (IPL) is the core of the H^3IoT. The data obtained from H^3IoT can be processed through the microcontroller or hardware platforms for further processing in upper layer. This network gateway attaches its information at IPL to top levels of 2G, 3G, 4G or Wi-Fi connectivity [28].

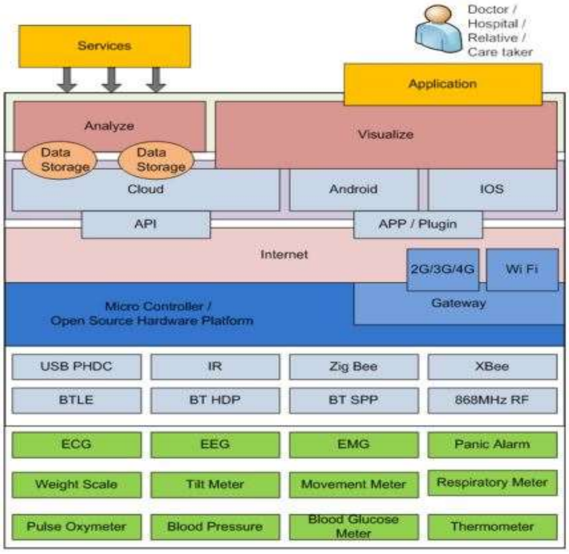


Figure (1): The Architecture for Home Health Hub IoT (H^3IoT) [28]

The User Application Layer (UAL) is H^3IoT top ingredients. As an approach to H^3IoT, UAL may be sent to doctors, parents, hospitals, and caregivers who collect information, for their counterpart in real-time, on the welfare of the elderly. The most important thing is to visualize IAL information received and processed in UAL. Also, stored cloud (IAL) details can be examined with the older individual's comprehensive medical history.

3.2 MBT for the Home Health Hub IoT

The MBT for the Home-Health Hub3IoT case study is built using an Eclipse-based open-source PCM-Bench platform. The PCM-Bench was used by software developers to build PCM meta-model instances and predict QoS analyses. All types of entities are described in PCM-Bench. Interfaces, data types, and components are PCM registry entities. The PCM supports the modeling of various component types to present different development stages and differentiate between simple and composite components on the component side of Hub^{3}IoT.

3.3 Approach of CBSM

Palladio is a visual programming language and a dynamic platform to simulate dynamic architecture designs like software. Our approach uses the Palladio Component Model (PCM) to specify component-based software architectures in a parametric way. For automated performance simulation, Palladio is designed to use models as an input that is as close as possible to regular architectural models, which should still be used when designing complex software systems. Palladio is a very well and widely used approach to predict quality properties of component-based software architectures.

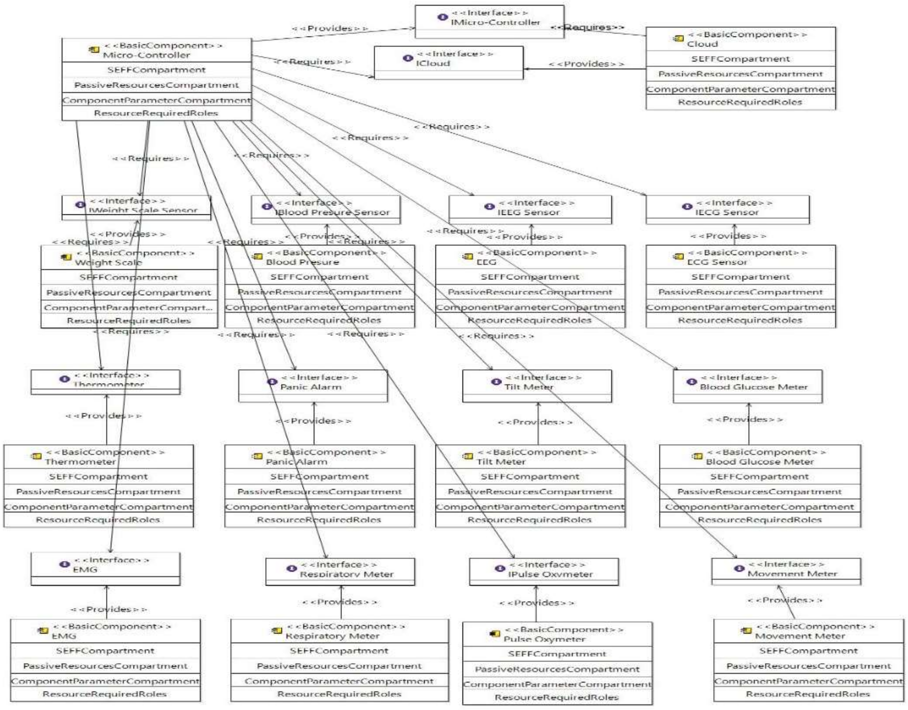
3.4 Component Repository Model of Hub3IoT

MBT basic concept is to construct complex repository software systems by incorporating the system basic components. Figure 2 depicts the Home Health Hub IoT case study CRM.

This category comprises components and interfaces. This will start by describing the entities that are placed in the repositories. After that, SEFF model component services abstract behavior and performance properties. Composite structures improve system predictability during the early design stages. On the other side, it helps the system designers to make changes. The modeling initial aim is to predict the quality of IoT applications before the consumer even realizes they exist.

3.4 Component Parameters Description

There are four basic components of Home Health Hub IoT. In the CRM of the H^{3}IoT each component has a specific resource demand according to its functionality in the system. The parameters description of these resources is presented in Table 1.



Figure(2): The Component Repository Model of Home Health Hub (H^{3}IoT)

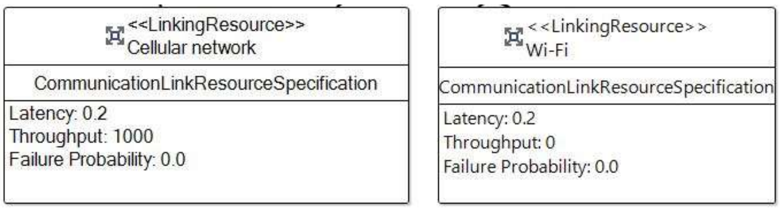
Table (1): Home Health Hub3IoT parameters description

Parameters	Description	Resources
P1	Internet network	Latency/Throughput
P2	Micro Controller	Number of Replicas/Processing Rate
P3	Cloud Processing Unit	Number of Replicas/Processing Rate
P4	Mobile Devices	Number of Replicas/Processing Rate

i. Internet network

Latency and throughput are the computing resources of a cellular or wireless network, as shown in Figure 3. Latency refers to the amount of time that the data requires to transfer from its source to the other side of the link. Throughput refers to the number of requests that a system can process per unit time.

Figure(3): Cellular network and Wireless network Component Model Linking Resource



<<LinkingResource>>
Cellular network
CommunicationLinkResourceSpecification
Latency: 0.2
Throughput: 1000
Failure Probability: 0.0

<<LinkingResource>>
Wi-Fi
CommunicationLinkResourceSpecification
Latency: 0.2
Throughput: 0
Failure Probability: 0.0

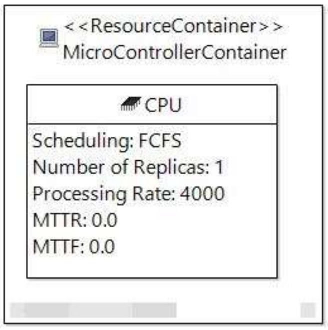
The values of the latency and throughput resources of the test model were based on the standard software values as demonstrated in Table 2.

Table (2): Resources values of Internet network

Parameters	latency	Throughput
P1.1	0.1	1000
P1.2	0.2	900
P1.3	0.4	800
P1.4	0.6	600
P1.5	0.8	400
P1.6	1	200

ii. Micro Controller

The CPU is a key demand for the Home Health Hub IoT, and it has two main effect resources, processing rate and replicas, as shown in Figure 4. The processing rate defines how often the resource will be used. The number of replicas attributed in the Palladio Processing Resource defines the total number of processors. The number of cores in the Home Health Hub IoT CPU is mapped to this attribution. The values of the processing rate and the number of replica resources of the test model were illustrated in Table 3.



<<ResourceContainer>>
MicroControllerContainer

CPU

Scheduling: FCFS
Number of Replicas: 1
Processing Rate: 4000
MTTR: 0.0
MTTF: 0.0

Figure (4): Home Health Hub3IoT Controller Resource Container

Table(3): Resources values of Home Health Hub3IoT controller CPU [29]

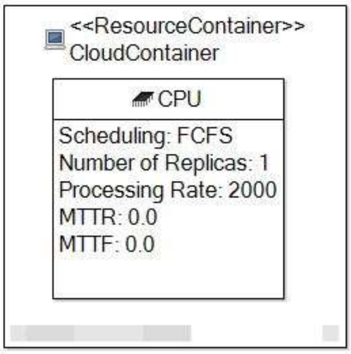
Parameters	Number of Replicas	Processing rate
P2.1	1	2000
P2.2	2	4000
P2.3	8	1000
P2.4	4	2000
P2.5	6	1000
P2.6	1	1000

iii. Cloud Processing Unit

The information would be processed in the cloud and analyzed for the elderly person’s detailed health records. The CPU is the most important demand in the Cloud, and it has two main effects on resources.

The processing rate and replica number, as shown in Figure 5. The processing rate defines how often the resource will be used. The number of replicas attributed in the Palladio Processing Resource categorizes the total number of processors.

This property is mapped to the Cloud CPU multiple cores. The values of the latency and throughput resources of the test model were based on the standard software values as demonstrated in Table 4 [29].


<<ResourceContainer>>
CloudContainer

CPU
Scheduling: FCFS
Number of Replicas: 1
Processing Rate: 2000
MTTR: 0.0
MTTF: 0.0

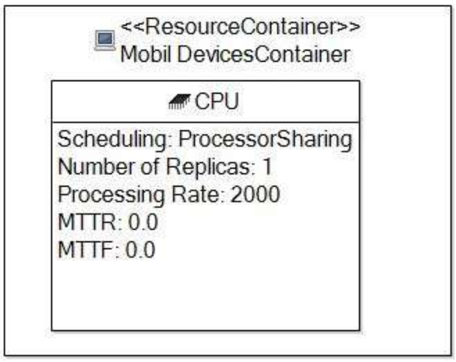
Figure (5): Home Health HubIoT Cloud Resource Container

Table 0): Resources values of Hub3IoT Cloud CPU [29]

Parameters	Number of Replicas	Processing rate
P3.1	1	2000
P3.2	2	4000
P3.3	8	1000
P3.4	4	2000
P3.5	6	1000
P3.6	1	1000

iv. Mobile Devices

A mobile app was created to improve the accessibility of the older person’s status in Home Health Hub IoT. It is a computer software program that can be used on smartphones, tablets, and other mobile devices. The processing rate and the number of replicas are the two key CPU resources that are affected by the mobile device controller resource container. The processing rate defines how much the resource can carry out the resource requirements. The number of processors in Palladio Processing Resource, on the other hand, is classified by the number of replicas attributed. As shown in Figure 6, this attribution represents the number of cores in a virtual Smart-Home controller CPU.



Figure(6):Home Health Hub^{3}IoT Mobil Devices ResourceContainer

The values of the processing rate and the number of replica resources of our test model were based on the standard software values. Table 5 illustrated these ranges [29].

Table (5): Resources values of Hub IoT Mobil Devices CPU [29]

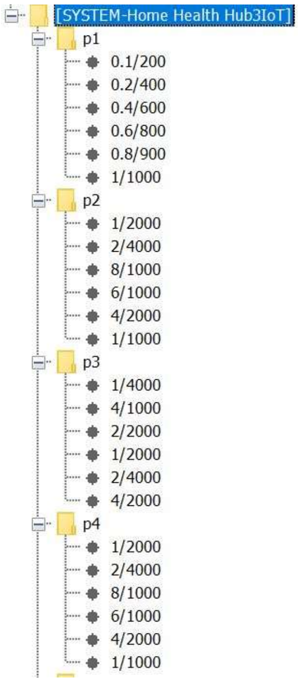
Parameters	**Number of Replicas**	**Processing rate**
P4.1	1	1000
P4.2	1	2000
P4.3	1	4000
P4.4	1	8000
P4.5	2	2000
P4.6	2	4000

3.5 The test cases of Home Healthy Hub3 IoT

There are four basic components in the Home Healthy Hub IoT repository diagram, each of them has specific resource consumption based on the system functionality. In order to generate possible test cases, the In-Parameter-Order-General (IPOG) algorithm was implemented and integrated into a combinatorial test generation tool which is Advanced Combinatorial Testing System (ACTS) [30] twice.

i. Without Constraints:

The results of the IPOG algorithm will be used to generate test cases, which will be run using the ACTS tool, as shown in Figure 7. The number of parameters used to represent ***n*** is four. For each of those parameters, there are 6 values that represent ***d***; the number of possible combinations for these parameters is d^n, which is 64 equal to 1,296 test case values. All these test cases are not attached here for the reason of space constraints, however they are available online Test cases of Healthy Hub3IoT MBTPA .



- [SYSTEM-Home Health Hub3IoT]
  - p1
    - 0.1/200
    - 0.2/400
    - 0.4/600
    - 0.6/800
    - 0.8/900
    - 1/1000
  - p2
    - 1/2000
    - 2/4000
    - 8/1000
    - 6/1000
    - 4/2000
    - 1/1000
  - p3
    - 1/4000
    - 4/1000
    - 2/2000
    - 1/2000
    - 2/4000
    - 4/2000
  - p4
    - 1/2000
    - 2/4000
    - 8/1000
    - 6/1000
    - 4/2000
    - 1/1000

Figure (7): Combinatorial Testing System (ACTS) structure of Home Healthy Hub^{3}IoT

ii. With Constraints:

It is always impractical to test all possible combinations exhaustively. As a consequence, a technique can only test a subset of possible combinations. We have constraints in place for integration strategies, which is a good approach. The constraints for placing together the combinations were set out. The combinations must satisfy the constraints in order to be valid. During the test generation process, the specified constraints will be taken into account, resulting in test cases that cover, and only cover, the combinations that conform to those constraints. The following are the constraints that were implemented:

- (p1<="MINIMUM Latency/ Throughput"), Throughput does not matter in our model.
- (p2<= Maximum "Number of Replicas/ Processing Rate"), to get MINIMUM response time.
- (p3<= Maximum "Number of Replicas/ Processing Rate"), to get MINIMUM response time.
- (p4<= Maximum. "Number of Replicas/ Processing Rate"), to get MINIMUM response time.

The same strategy was used as in approach one (without constraint). Using the ACTS tool, we developed and ran our system by implementing the IPOG algorithm and applying the mentioned constraints. As a result, we got a sub-set containing 40 test cases. Table 6shows the configurations of a subset of test cases.

Table (6): Sub-set of test cases of Home Healthy Hub[3] IoT generated by ACTS tool

 # ACTS Test Suite Generation

 # Degree of interaction coverage: 4
 # Number of parameters: 4
 # Maximum number of values per parameter: 6
 # Number of configurations: 40

Test Case No.	p1	p2	p3	p4
1	0.1/200	1/2000	1/4000	1/2000
2	0.1/200	1/2000	1/4000	1/1000
3	0.1/200	1/2000	4/1000	1/2000
4	0.1/200	1/2000	4/1000	1/1000
5	0.1/200	1/2000	2/2000	1/2000
6	0.1/200	1/2000	2/2000	1/1000
7	0.1/200	1/2000	1/2000	1/2000
8	0.1/200	1/2000	1/2000	1/1000
9	0.1/200	1/2000	2/4000	1/2000

 # ACTS Test Suite Generation

 # Degree of interaction coverage: 4
 # Number of parameters: 4
 # Maximum number of values per parameter: 6
 # Number of configurations: 40

Test Case No.	p1	p2	p3	p4
10	0.1/200	1/2000	2/4000	1/1000
11	0.1/200	1/1000	1/4000	1/2000
12	0.1/200	1/1000	1/4000	1/1000
13	0.1/200	1/1000	4/1000	1/2000
14	0.1/200	1/1000	4/1000	1/1000
15	0.1/200	1/1000	2/2000	1/2000
16	0.1/200	1/1000	2/2000	1/1000
17	0.1/200	1/1000	1/2000	1/2000
18	0.1/200	1/1000	1/2000	1/1000
19	0.1/200	1/1000	2/4000	1/2000
20	0.1/200	1/1000	2/4000	1/1000
21	0.2/400	1/2000	1/4000	1/2000
22	0.2/400	1/2000	1/4000	1/1000
23	0.2/400	1/2000	4/1000	1/2000
24	0.2/400	1/2000	4/1000	1/1000
25	0.2/400	1/2000	2/2000	1/2000
26	0.2/400	1/2000	2/2000	1/1000
27	0.2/400	1/2000	1/2000	1/2000
28	0.2/400	1/2000	1/2000	1/1000
29	0.2/400	1/2000	2/4000	1/2000
30	0.2/400	1/2000	2/4000	1/1000
31	0.2/400	1/1000	1/4000	1/2000
32	0.2/400	1/1000	1/4000	1/1000
33	0.2/400	1/1000	4/1000	1/2000
34	0.2/400	1/1000	4/1000	1/1000
35	0.2/400	1/1000	2/2000	1/2000
36	0.2/400	1/1000	2/2000	1/1000
37	0.2/400	1/1000	1/2000	1/2000
38	0.2/400	1/1000	1/2000	1/1000
39	0.2/400	1/1000	2/4000	1/2000
40	0.2/400	1/1000	2/4000	1/1000

4 Results of Home Healthy Hub H3IoT MBTPA

There are 40 test cases representing the best test cases concerning performance quality factors after applying the constraints to all possible test cases and running them through the ACTS tool in chapter five. PCM used these cases to assess the Home Healthy Hub IoT performance metrics. We chose the best case out of the 40 to see how it did. Two main factors assess the importance of success in MBTPA: response time and resource utilization. The 40 cases were run in five-case groups to make it easier to notice each case's behavior and choose the best response time and resource allocation.

i. Test cases 1-5

Cases 1–5 will now be addressed in terms of response times and resource use (see table 6). The response times for cases 1 through 5 are shown in Figure 8.

The third case, depicted in green, is clearly the best. The system has a 97 percent chance of responding to a call in less than 0.4 milliseconds in this situation, which is considered a reasonable response time for Healthy-Home applications. Figure 9 depicts the test cases 1-5 from the perspective of the resource utilization metric. In these five test cases, the CPU resource demands are nearly identical. In this scenario, the main CPU had been running at 80% for the previous 3.5 milliseconds. As a result, the amount of hardware resources 3.5 milliseconds. As a result, the amount of hardware resources used per time unit is kept to a minimum.

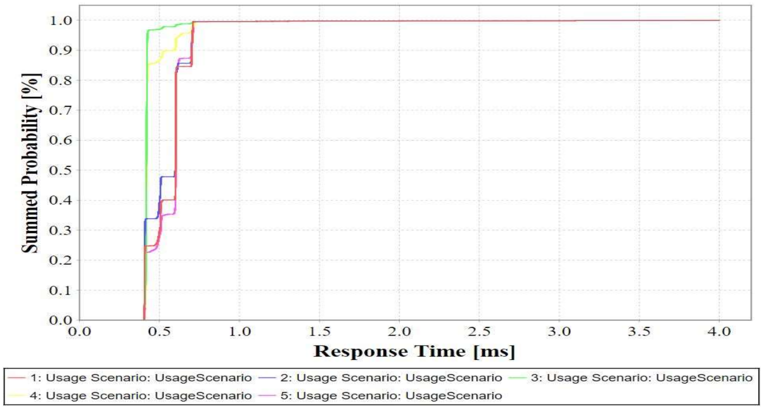


Figure (8): The Cumulative Distribution Function (CDF) of Response Time Tuple for cases 1-5

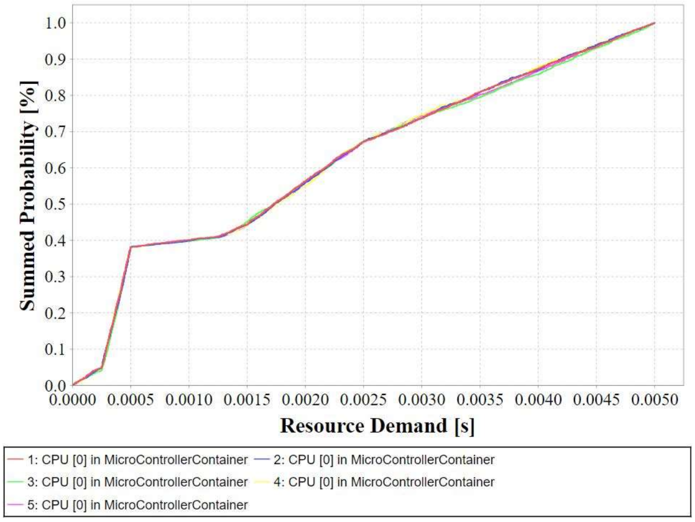


Figure (9): The Cumulative Distribution Function (CDF) of CPU in Microcontroller of cases 1-5

ii. Test cases 6-10

Response times and resource utilization will now be analyzed in cases 6–10. (See table 6).

In all these test cases the system has approximately 90% possibility of responding to a call in less than 1.5 milliseconds as shown in Figure 10.

Figure 11, depicts the test cases 6-10, from the perspective of the resource utilization metric. In all test cases (6-10), the CPU resource demands are nearly equal. In this scenario, the main CPU had been running at approximately 80% for the previous 3.5 milliseconds. As a result, the amount of hardware resources used per time unit is kept to a minimum

.

Figure (10): The Cumulative Distribution Function (CDF) of Response Time Tuple for cases 6-10

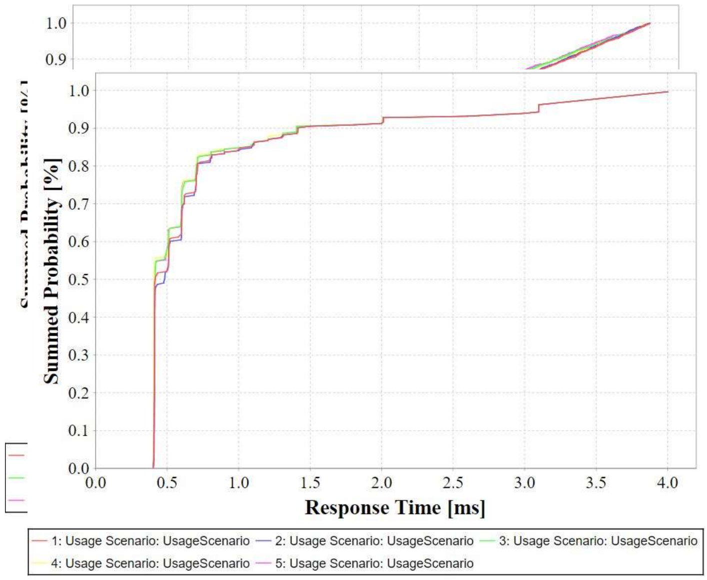


Figure (11): The Cumulative Distribution Function (CDF) of CPU in Microcontroller of cases 6-10

iii. Test cases 11-15

Response times and resource usage will be discussed in test cases 11–15. (See Table 6 for more information.) In these test cases, the system has a 90% probability of responding to a call in about 2 milliseconds, as shown in Figure 12, which is an adequate but not ideal response time for Healthy-Home applications. The test cases 11-15, are represented in Figure 13 from the perspective of the resource utilization metric. The CPU resource demands in all of these test cases are approximately identical. The main CPU had been running at 80% for the previous 7 milliseconds.

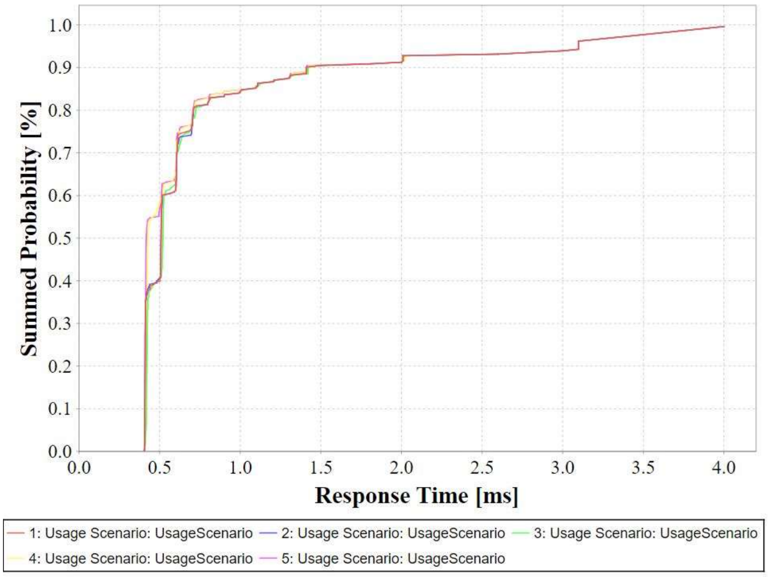


Figure (12): The Cumulative Distribution Function (CDF) of Response Time Tuple for cases 11-15

The test cases 11-15, are represented in Figure 13 from the perspective of the resource utilization metric. The CPU resource demands in all of these test cases are approximately identical. The main CPU had been running at 80% for the previous 7 milliseconds.

i. Test cases 16-20

In cases 16–20, response times and resource usage will be examined. (For further information, see Table 6). In these test cases, 16-20, the system has a 90% probability of responding to a call in less than 1.5 milliseconds, as shown in Figure 14, which is an adequate but not ideal response time for Healthy-Home applications in comparison to other test cases.

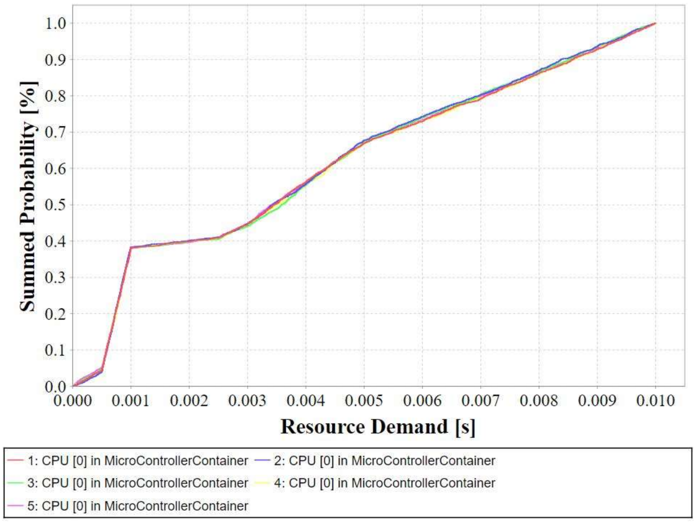


Figure (13): The Cumulative Distribution Function (CDF) of CPU in Microcontroller of cases 11-15

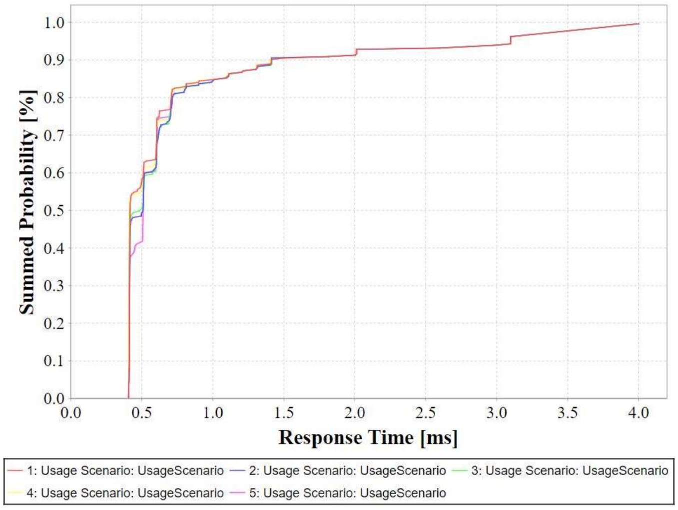


Figure (14): The Cumulative Distribution Function (CDF) of Response Time Tuple for cases 16-20

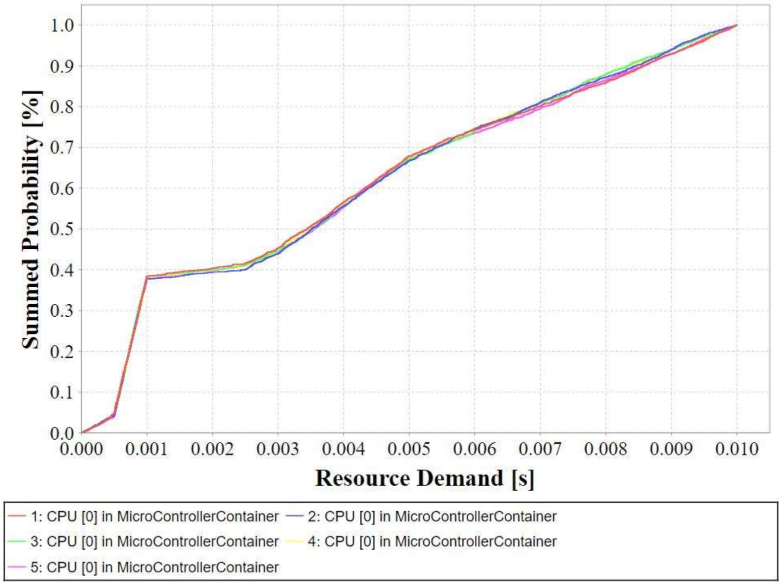


Figure (15): The Cumulative Distribution Function (CDF) of CPU in Microcontroller of cases 16-20

Figure 15, depicts the test cases 16-20 from the perspective of the resource usage metric. In all of these test cases, the CPU resource demands are nearly equal. For the previous 8 milliseconds, the main CPU had been running at about 90%.

ii. Test cases 21-25

Response times and resource use will be discussed in cases 21–25. (See Table 6 for more information.). According to Figure 16, nearly all of the test cases 21-25, had the same response time. The system has just a 20 percent probability of responding to a call in about 3 milliseconds, which is considered a long response time.

Figure 17, represents the test cases 21-25 from the perspective of the resource usage metric. In both of these test cases, the CPU resource demands are approximately similar. For the previous 4 milliseconds, the main CPU had been operating at 90%. As a result, the amount of hardware resources used per time unit is kept to a bare minimum.

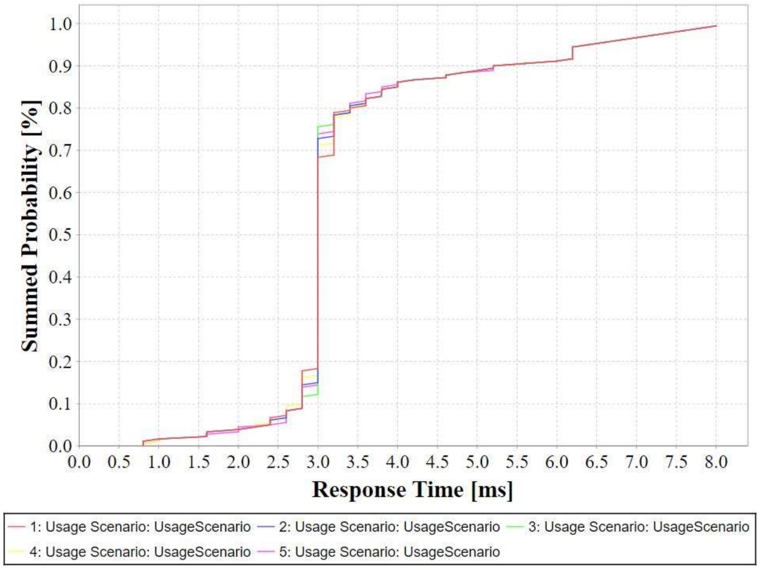


Figure (16): The Cumulative Distribution Function (CDF) of Response Time Tuple for cases 21-25

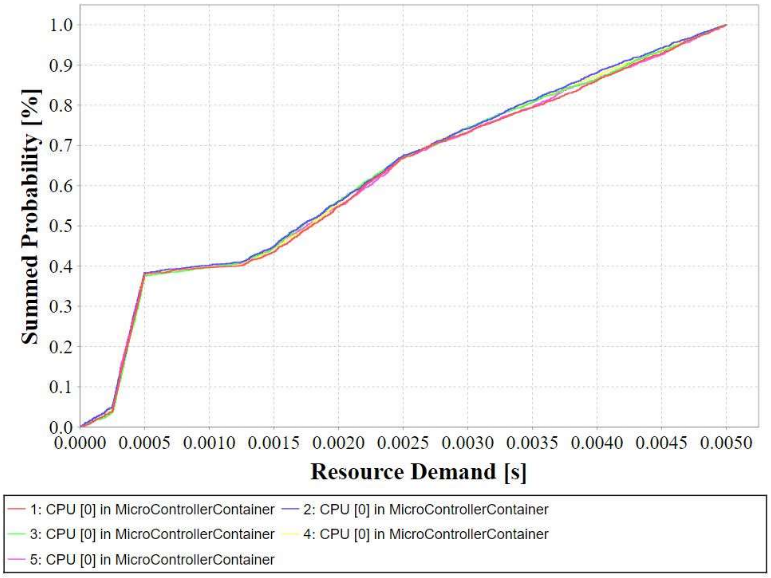


Figure (17): The Cumulative Distribution Function (CDF) of CPU in Microcontroller of cases 21-25

iii. Test cases 26-30

In cases 26–30, response times and resource usage will be examined. (For further information, see Table 6). Figure 18 indicates that nearly all of the test cases, 26-30, had the same response time. The system has a 70 percent chance of responding to a call in around 3 milliseconds.

Figure 19 represents the test cases 26-30, from the perspective of the resource usage metric. In all of these test cases, the CPU resource demands are almost similar. For the previous 4 milliseconds, the main CPU had been operating at 90%.

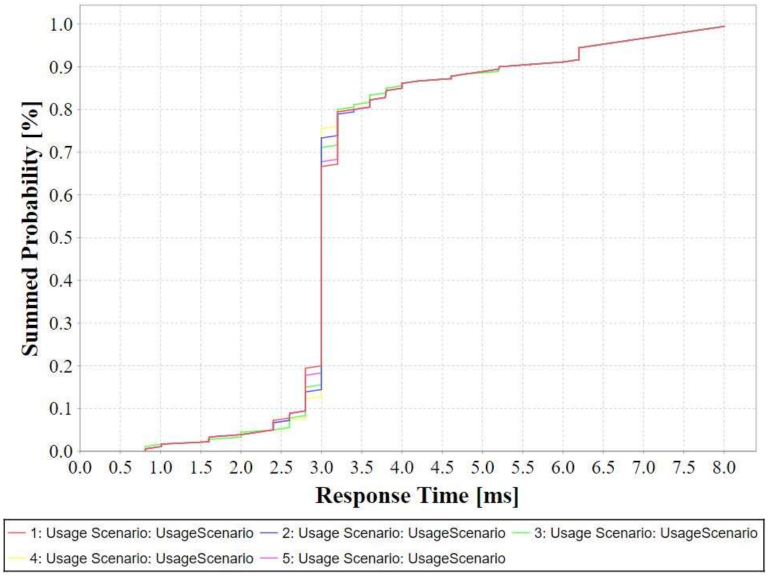


Figure (18): The Cumulative Distribution Function (CDF) of Response Time Tuple for cases 26-30

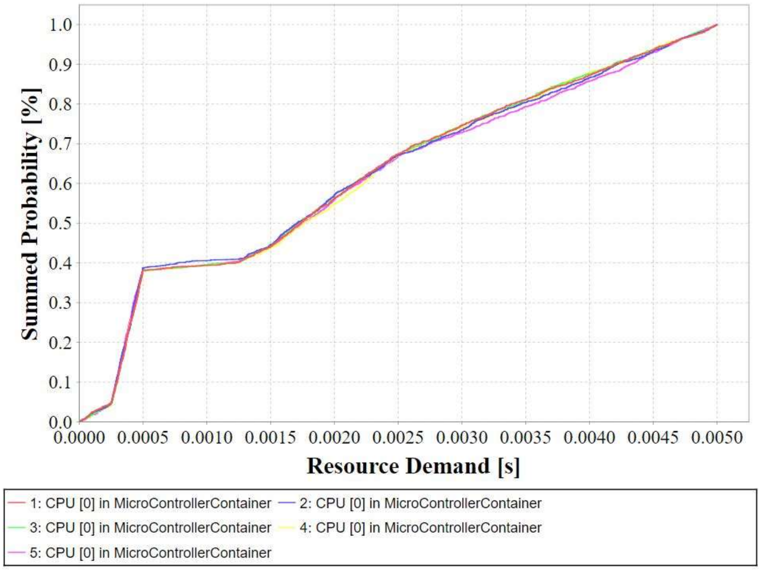


Figure (19): The Cumulative Distribution Function (CDF) of CPU in Microcontroller of cases 26-30

iv. Test cases 31-35

Cases 31-35 will now look at response times and resource usage. (For more information, see table 6). Nearly all of the test cases 31-35, had the same response time, as shown in Figure 20 In around 3 milliseconds, the system has a 75% chance of responding to a call.

From the perspective of the resource consumption metric, Figure 21 depicts the test cases 31-35. The CPU resource demands in all of these test cases are nearly identical. The main CPU had been running at 89 percent for the previous 8 milliseconds.

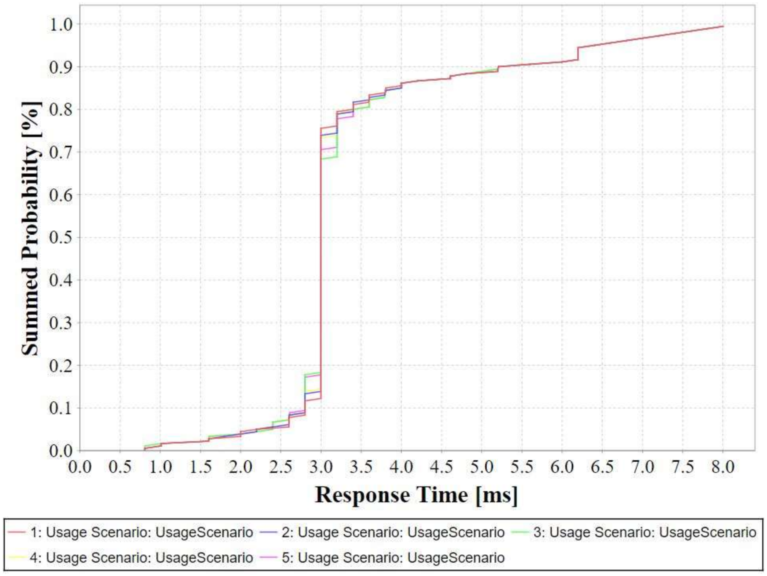


Figure (20): The Cumulative Distribution Function (CDF) of Response Time Tuple for cases 31-35

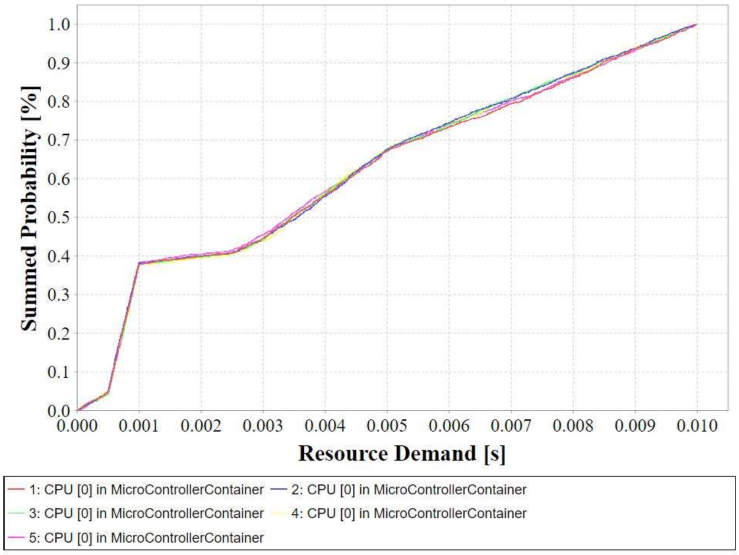


Figure (21): The Cumulative Distribution Function (CDF) of CPU in Microcontroller of cases 31-35

v. Test cases 36-40

Response times and resource use will be investigated in cases 36–40. (See Table 6 for more information.)

Approximately all of the test cases 36-40, had the same response time, as shown in Figure 22. In about 3 milliseconds, the system has a 75% chance of responding to a call. The test cases 36-40, are represented in Figure 23 from the perspective of the resource utilization metric. The CPU resource demands are approximately identical in all of the test cases (36-40). The main CPU had been operating around 65% for the previous 5 milliseconds in these cases.

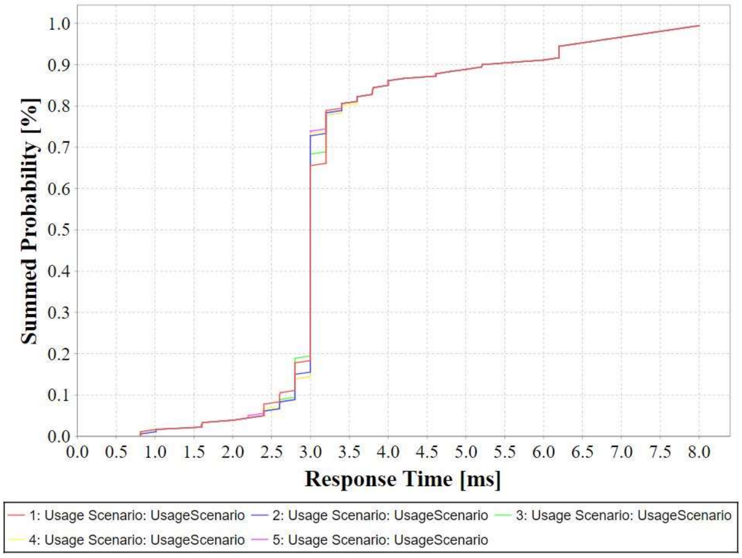


Figure (22): The Cumulative Distribution Function (CDF) of Response Time Tuple for cases 36-40

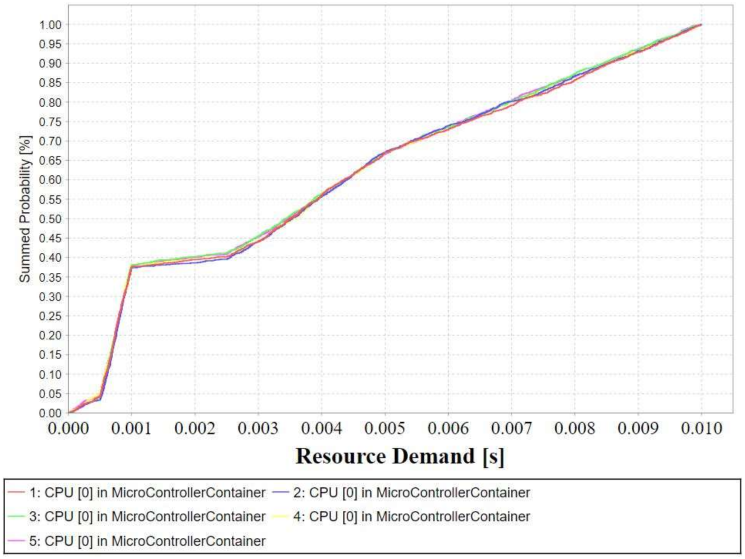


Figure (23): The Cumulative Distribution Function (CDF) of CPU in Microcontroller of cases 36-40

After analyzing all 40 test cases, we concluded that the test cases that were the most successful in terms of response time and resource utilization were 1-20. Each of them is an ideal test case with the best output performance metrics.

As a result, the latency (P1) is the most effected factor in the behavior of these test cases, since in test cases 1-20, we used the lower latency value of 0.1 and throughput was not necessary in this system.

5 Conclusion

In this research study, a new technique (we named it MBTPA) has been proposed, designed and simulated for three cases using a dynamic visual programming platform, Palladio. The main conclusions are:

1. The proposed MBTPA model was employed to assess three cases of different levels of IoT applications. MBTPA has overcome one of the most common IoT testing challenges: system heterogeneity.
2. The Home Health Hub IoT was used as a third case study to implement the methodological approach into practice. The ACTS tool was used to generate the

possible test cases, as it had been in previous cases. Each test case's performance metrics were monitored until we identified the test cases that were the most efficient in terms of response time and resource utilization (1-20). They all are perfect test cases with the highest production efficiency indicators. As a consequence, the latency (P1) is the most important factor in the conduct of these test cases, as we used the lower latency value of 0.1 in test cases 1-20.

3. Testers are now able to use MBTPA before the implementation, allowing designers to take the time and resources they need to complete their IoT project without worrying about wasting it.

6 Future work

Three separate case studies have been used to plan and execute the current work. To run the MBTPA model, the PCM was used. The PCM focuses on the factors that influence the system quality properties. We designed and ran the MBTPA for our analysis, and we got good results that we can trust. As a result of our work, we recommend the following points for future research.

1. Designing and implementing MBTRA as a Model-Based Testing for Reliability Attribute.
2. Establish a standard process and logo to show the IoT certification test.
3. Develop research methods for IoT applications testing.
4. Assess the throughput measure using our model (MBTPA).

7 References:

[1] M. H. Qasem and W. AlMobaideen, "Heterogeneity in IoT-based Smart Cities Designs," *Int. J. Interact. Mob. Technol.*, vol. 13, no. 12, 2019.

[2] S. Lam and W. Chung, "Uses of Internet and Mobile Technology in Health Systems for the Elderly.," *Int. J. Interact. Mob. Technol.*, vol. 4, no. 2, 2010.

[3] D. Nirmala and T. LathaMaheswari, "Automated testcase generation for software quality assurance," in *2016 10th International Conference on Intelligent Systems and Control (ISCO)*, 2016, pp. 1–6.

[4] M. Fahmideh and D. Zowghi, "An exploration of IoT platform development," *Inf. Syst.*, vol. 87, p. 101409, 2020.

[5] Z. Ming and M. A. Yan, "QoS-aware computational method for IoT composite service," *J. China Univ. Posts Telecommun.*, vol. 20, pp. 35–39, 2013.

[6] Y. Berhanu, H. Abie, and M. Hamdi, "A testbed for adaptive security for IoT in eHealth," in *Proceedings of the International Workshop on Adaptive Security*, 2013, pp. 1–8.

[7] L. Li, M. Rong, and G. Zhang, "An Internet of Things QoE evaluation method based on multiple linear regression analysis," in *2015 10th International Conference on Computer Science & Education (ICCSE)*, 2015, pp. 925–928.

[8] Q. Mateen and M. Sirshar, "Software QualityAssurance in Internet of Things," *Int. J. Comput. Appl*, vol. 109, no. 9, pp. 16–24, 2015.

[9] H. Foidl and M. Felderer, "Data science challenges to improve quality assurance of Internet of Things applications," in *International Symposium on Leveraging Applications of Formal Methods*, 2016, pp. 707–726.

[10] T. Banerjee and A. Sheth, "Iot quality control for data and application needs," *IEEE Intell. Syst.*, vol. 32, no. 2, pp. 68–73, 2017.

[11] B. Amrutur *et al.*, "An open smart city IoT test bed: street light poles as smart city spines," in *Proceedings of the Second International Conference on Internet-of-Things Design and Implementation*, 2017, pp. 323–324.

[12] K. K. Nair, E. Dube, and S. Lefophane, "Modelling an IoT testbed in context with the security vulnerabilities of South Africa," in *2017 3rd IEEE International Conference on Computer and Communications (ICCC)*, 2017, pp. 244–248.

[13] S. Li and J. Huang, "GSPN-based reliability-aware performance evaluation of IoT services," in *2017 IEEE International Conference on Services Computing (SCC)*, 2017, pp. 483–486.

[14] M. Tappler, B. K. Aichernig, and R. Bloem, "Model-based testing IoT communication via active automata learning," in *2017 IEEE International conference on software testing, verification and validation (ICST)*, 2017, pp. 276–287.

[15] G. White, V. Nallur, and S. Clarke, "Quality of service approaches in IoT: A systematic mapping," *J. Syst. Softw.*, vol. 132, pp. 186–203, 2017.

[16] Z. Wang, X. Dong, Y. Li, L. Fang, and P. Chen, “IoT security model and performance evaluation: a blockchain approach,” in *2018 International Conference on Network Infrastructure and Digital Content (IC-NIDC)*, 2018, pp. 260–264.

[17] Y. Liu, K. Zheng, P. Craig, Y. Li, Y. Luo, and X. Huang, “Evaluating the Reliability of Blockchain Based Internet of Things Applications,” in *2018 1st IEEE International Conference on Hot Information-Centric Networking (HotICN)*, 2018, pp. 230–231.

[18] T. Kuroiwa, Y. Aoyama, and N. Kushiro, “A hybrid testing environment between execution test and model checking for IoT system,” in *2019 IEEE International Conference on Consumer Electronics (ICCE)*, 2019, pp. 1–2.

[19] M. R. Nakhkash, T. N. Gia, I. Azimi, A. Anzanpour, A. M. Rahmani, and P. Liljeberg, “Analysis of performance and energy consumption of wearable devices and mobile gateways in iot applications,” in *Proceedings of the International Conference on Omni-Layer Intelligent Systems*, 2019, pp. 68–73.

[20] S. Siboni *et al.*, “Security testbed for internet-of-things devices,” *IEEE Trans. Reliab.*, vol. 68, no. 1, pp. 23–44, 2019.

[21] B. S. Ahmed, M. Bures, K. Frajtak, and T. Cerny, “Aspects of quality in Internet of Things (IoT) solutions: A systematic mapping study,” *IEEE Access*, vol. 7, pp. 13758–13780, 2019.

[22] A. Lepekhin, A. Borremans, I. Ilin, and S. Jantunen, “A systematic mapping study on internet of things challenges,” in *2019 IEEE/ACM 1st International Workshop on Software Engineering Research & Practices for the Internet of Things (SERP4IoT)*, 2019, pp. 9–16.

[23] R. Jabbar, M. Krichen, M. Kharbeche, N. Fetais, and K. Barkaoui, “A Model-Based Testing Framework for Validating an IoT Solution for Blockchain-Based Vehicles Communication,” 2020.

[24] M. Bures, M. Macik, B. S. Ahmed, V. Rechtberger, and P. Slavik, “Testing the Usability and Accessibility of Smart TV Applications Using an Automated Model-based Approach,” *IEEE Trans. Consum. Electron.*, vol. 66, no. 2, pp. 134–143, 2020.

[25] I. I. Hamarash, “MODEL-Based Performance Quality Assessment for IoT Applications.,” *Int. J. Interact. Mob. Technol.*, vol. 15, no. 12, 2021.

[26] I. I. Teeba and Hamarash, “Model-Based Quality Assessment of Internet of Things Software Applications: A Systematic Mapping Study.,” *Int. J. Interact. Mob. Technol.*, vol. 14, no. 9, 2020.

[27] I. O. f. Standardization, “The International Organization for Standardization (ISO),” *ISO*, 2019. https://www.iso.org/home.html

[28] P. P. Ray, “Home Health Hub Internet of Things (H 3 IoT): An architectural framework for monitoring health of elderly people,” in *2014 International Conference on Science Engineering and Management Research (ICSEMR)*, 2014, pp. 1–3.

[29] S. Becker, H. Koziolek, and R. Reussner, “Model-based performance prediction with

the palladio component model," in *Proceedings of the 6th international workshop on Software and performance*, 2007, pp. 54–65.

[30] L. Yu, Y. Lei, M. Nourozborazjany, R. N. Kacker, and D. R. Kuhn, "An efficient algorithm for constraint handling in combinatorial test generation," in *2013 IEEE Sixth International Conference on Software Testing, Verification and Validation*, 2013, pp. 242–251.

هەڵسەنگاندنی کوالیتی ئینتەرنێتی شتەکان (IoT) بەکارهێنانە پزیشکییەکان

پوختە:

ژمارەی ئەو ئەپڵیکەیشنانەی کە چەمکی ئینتەرنێتی شتەکان (IoT) بەکاردەهێنن، بە خێرایی زیاد دەکات. ئەم گەشەکردنە بەبێ دڵنیایی کوالیتی بەرز ناتوانێت بەردەوام بێت. بەپێی ئەدەبیاتی پێشوو، پێویستییەکی بەپەلە هەیە بۆ دروستکردنی مۆدێلێکی تایبەتی دڵنیایی کوالیتی (QA) بۆ بەرنامەکانی IoT. لە ئەنجامدا زۆر ئەستەمە یەک مۆدێلی دڵنیایی کوالیتی بۆ هەموو ئەم بەرنامانە هەبێت. پێداچوونەوەی تەواوی ئەدەبیاتی توێژینەوەکە دەتوانرێت لەم لینکەدا بدۆزرێتەوە https://docs.google.com/spreadsheets/SLSIoT. نایەکسانی سیستەم، تێچووی تاقیکردنەوە، کات کە بۆ تاقیکردنەوەی سیستەمەکە وەرگیراوە، و وردی ئەنجامی تاقیکردنەوەکان هەموویان تەحەددداین لە تاقیکردنەوەی بەرنامەکانی IoT.

تاقیکردنەوە لەسەر بنەمای مۆدێل کە بە باشی ناسراوە و ڕەنگە باشترین چارەسەر بێت بۆ چارەسەرکردنی ئەم کێشانە، بریتییە لە مۆدێلکردنی سیستەم بۆ مەبەستی هەڵسەنگاندن (MBT). ئامانجی ئەم توێژینەوەیە پەرەپێدانی شێوازێکی تاقیکردنەوە لەسەر بنەمای مۆدێلە بۆ هەڵسەنگاندنی کوالیتی ئەدای بەرنامەکانی IoT.

تەندروستی ماڵەوە Hub3IoT وەک توێژینەوەیەکی کەیس بەکارهات بۆ پەرەپێدانی ڕێبازی میتۆدۆلۆژی Hub3IoT پارادایمێکی نوێیە لەسەر بنەمای IoT کە هەستەوەری زیندوو، تۆڕەکانی پەیوەندی، مایکرۆکۆنترۆڵەر، دەروازەکان، ئینتەرنێت، بەرنامەکان و پەیوەندییە نێوانیان لەخۆدەگرێت. ئامرازی بنیاتنەری تاقیکردنەوەی تێکەڵاو (ACTS) بەکارهات بۆ دروستکردنی کەیسی تاقیکردنەوەی ئەگەری. پێوەرەکانی ئەدای هەر حاڵەتێکی تاقیکردنەوە بەدواداچوونیان بۆ کرا تا

باشترین کەیسی تاقیکردنەوە دۆزرایەوە، کە خێراترین کاتی وەڵامدانەوەی هەبوو کە هۆکارێکی گرنگە لەم بەرنامە ناسکانەدا و کەمترین بڕی سەرچاوەی بەکارهێنراو. بەپێی ئەنجامەکان، شێوازی تاقیکردنەوەی پێشنیارکراوی ورد دەرکەوتووە.

تقييم جودة التطبيقات الطبية لإنترنت الأشياء (IoT)

الملخص:

يتزايد عدد التطبيقات التي تستخدم مفهوم إنترنت الأشياء (IoT) بشكل سريع. لا يمكن أن يستمر هذا النمو بدون ضمان الجودة العالية. وفقًا للأدبيات السابقة، هناك حاجة ماسة إلى بناء نموذج لضمان الجودة (QA) خاص لتطبيقات إنترنت الأشياء. نتيجة لذلك، من الصعب للغاية الحصول على نموذج واحد لضمان الجودة لجميع هذه التطبيقات. يمكن العثور على مراجعة الأدبيات الخاصة بالبحث بصورة كاملة على https://docs.google.com/spreadsheets/SLSIoT. يعد عدم تجانس النظام ونفقات الاختبار والوقت المستغرق لاختبار النظام ونسبة دقة نتائج الاختبار كلها تحديات في اختبار تطبيقات إنترنت الأشياء.
يتضمن الاختبار المستند إلى النموذج ، المعروف جيدًا وربما الحل الأفضل لحل هذه المشكلات ، نمذجة النظام لأغراض التقييم (MBT). الهدف من هذه الدراسة هو تطوير طريقة اختبار قائمة على النموذج لتقييم جودة أداء تطبيقات إنترنت الأشياء.
تم استخدام Home Health Hub^{3}IoT كدراسة حالة لوضع النهج المنهجي Hub^{3}IoT هو نموذج جديد قائم على إنترنت الأشياء يتضمن المستشعرات الحيوية وشبكات الاتصال والميكرو كونترولر والبوابات والإنترنت والتطبيقات بالإضافة إلى ترابطها معاً. تم استخدام أداة إنشاء الاختبار التجميعي(ACTS) لإنشاء حالات الاختبار المحتملة. وكذلك تم تتبع مقاييس أداء كل حالة اختبار حتى تم العثور على أفضل حالة اختبار ،والتي كانت تتمتع بأسرع وقت استجابة والذي يعتبر عامل مهم في هذه التطبيقات الدقيقة وكذلك تم العثور على أقل قدر من الموارد المستخدمة. وفقًا للنتائج، وجد ان طريقة الاختبار المقترحة دقيقة.