

Research Article

A Multidimensional Internet of Things Testbed System: Development and Evaluation

Ibrahim S. Alsukayti 

Department of Computer Science, College of Computer, Qassim University, Buraydah, Saudi Arabia

Correspondence should be addressed to Ibrahim S. Alsukayti; skiety@qu.edu.sa

Received 4 August 2020; Revised 5 September 2020; Accepted 25 September 2020; Published 14 October 2020

Academic Editor: Shuhui Yang

Copyright © 2020 Ibrahim S. Alsukayti. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The technological breakthrough of the Internet of Things (IoT) drives the emergence of a wide scope of smart IoT solutions in different domains. Advancing the different technological aspects of these solutions requires effective IoT implementations and experimentations. This is widely addressed following low-cost and scalable methods such as analytical modeling and simulation. However, such methods are limited in capturing physical characteristics and network conditions in a realistic manner. Therefore, this paper presents an innovative IoT testbed system which facilitates practical experimentation of different IoT solutions in an effective environment. The testbed design was developed towards a general-purpose multidimensional support of different IoT properties including sensing, communication, gateway, energy management, data processing, and security. The implementation of the testbed was realized based on integrating a set of robust hardware components and developing a number of software modules. To illustrate its effectiveness, the testbed was utilized to experiment with energy efficiency of selected IoT communication technologies. This resulted in lower energy consumption using the Bluetooth Low Energy (BLE) technology compared to the Zigbee and 6LoWPAN technologies. A further evaluation study of the system was carried out following the Technology Acceptance Model (TAM). As the study results indicated, the system provides a simple yet efficient platform for conducting practical IoT experiments. It also had positive impact on users' behavior and attitude toward IoT experimentation.

1. Introduction

Internet of Things (IoT) is a technological move towards effective convergence of the physical and digital worlds. It starts to significantly influence our daily lives as it finds its way into different governmental, industrial, and commercial domains. Nowadays, there is a growing interest in IoT development considering different applications such as smart home, e-healthcare, and intelligent transportation systems. Real deployment of IoT systems becomes widespread in different regions of the globe. The number of IoT-enabled objects is increasing to reach high figures in the near future.

Along with the rapid development of the IoT technology, wide range of IoT solutions has emerged. On the other hand, advancing the different technological aspects of IoT becomes a trend research direction. Different research questions have been raised up in a broad array of IoT applications. These are related to different aspects such as which and how IoT communication technologies should be set up, what and how IoT

data should be collected and processed, and where IoT nodes should be positioned. However, such considerations critically require effective testing and experimentation.

There are a number of methods that can be adopted for this process. The most common ones are analytical modeling, simulation, and testbed experimentation. Analytical modeling relies on mathematical and statistical methods, whereas simulation is widely adopted as a low-cost and flexible experimentation approach. However, these methods have well-known limitations as they rely on modeling physical characteristics and reflecting physical phenomena using approximations and simplifications. This makes it complex to capture all natural hardware characteristics and realistic network conditions. It is challenging for simulation software to efficiently well-handle imperfections in radio communication, hardware interaction, sensing, and network traffic. Therefore, the need for IoT implementation and evaluation in a natural and real environment is evident. Having experimental IoT setups developed using physical IoT components

with real IoT data traffic is a more effective experimentation approach. Although it comes with a cost in terms of money and time, building IoT testbed systems would allow increasing the realism of testing environment and improving the credibility of the evaluation process.

On the other hand, building a motivating academic and research environment is important for advancing the process to a further limit. This is one of the main strategic objectives towards which the College of Computer in Qassim University strives. Accordingly, a number of academic resources have been made available and a variety of research activities has been supported. However, there are still no customized facilities available to support IoT research and education in the college. Researchers usually have to build their own experimental IoT setups for carrying out certain implementation and testing procedures. Educators have limited resources to acquire hands-on experience and interactive learning when it comes to practical IoT concepts. Given the growing interest in IoT technology, having a general-purpose and open-access IoT testbed would provide collaborative research environment and interactive learning platform. It can be, for example, utilized to investigate IoT communication technologies, study IoT network performance, test IoT security techniques, prototype IoT applications, and analyze IoT data. Developing an IoT testbed is envisaged to open the door for different opportunities including multidisciplinary cooperation. It is a cost-effective and time-saving approach to effectively promote IoT research and education.

Such a challenge is addressed in this paper which presents the design and implementation of a practical testbed system for IoT experimentation. It represents a step forward towards moving IoT experimentation to a more realistic and practical level. The main contribution of the proposed testbed system is the novel multidimensional support providing the following:

- (i) An IoT experimentation facility for effective testing of different IoT aspects in a realistic and controlled environment. It can be practically utilized to experiment with IoT networking considering a number of IoT communication protocols. It also supports IoT experimentation targeting IoT data processing, gateway operations, cloud integration, node deployment, and security
- (ii) An educational platform for practical and interactive IoT teaching-learning process. It can be easily implemented for practical teaching of different IoT concepts and technologies in addition to enabling interactive and practice-based IoT education

The testbed also features a novel combination of a number of characteristics including the following:

- (i) General-purpose multidimensional support of different IoT properties (sensing, communication, gateway, energy management, data processing, and security)

- (ii) Scalable, portable, and simple system architecture
- (iii) Cost-effective design and implementation using Commercial Off-The-Shelf (COTS) components
- (iv) Effective support of a wide range of IoT communication technologies (BLE, Zigbee, 6LoWPAN, and LoRaWAN)
- (v) Easy integration of different sensor devices
- (vi) Cloud integration
- (vii) Effective IoT data visualization
- (viii) Flexible access to the testbed using a web interface and API calls
- (ix) Easy implementation of a wide range of IoT applications
- (x) Support of node mobility

The testbed was built based on effective design considering high levels of simplicity, scalability, usability, portability, and cost-effectiveness. The main focus was on the realization of a ready-to-use and easy-to-control multidimensional testbed for time-saving and cost-cutting IoT experimentation. The testbed implementation was realized based on integrating a set of well-known hardware components and developing a number of software modules. The efficiency of the testbed to implement and run different IoT experiments is illustrated in this paper by a simple IoT use case. The evaluation of the system was conducted using the Technology Acceptance Model (TAM). As the results indicated, the system provides simple-to-use and efficient facility for effective IoT testing. It is also evident that the system succeeded in promoting positive attitudes towards practical IoT experimentation and attractive facility for future IoT testing.

In the following section, a summary of the related work is provided. Section 3 presents a technical overview of the IoT technology is presented. In Section 4, the main requirements of the proposed testbed system are discussed. Sections 5 and 6 present the system design and implementation of the proposed IoT testbed, respectively. Section 7 illustrates some example use cases of the testbed. A description of the system evaluation and discussion of the results are presented in Section 8. Section 9 is the conclusion to this paper.

2. Related Work

The research community has realized the evolving IoT technology and recognized the need for effective IoT experimentation. As indicated in [1], it is a common research practice for the simulation-based IoT studies to be validated with physical testbed experimentation. There have been a number of research efforts made to develop and deploy different physical IoT experimentation testbeds. Most of these testbeds were made publically available and openly accessible with varying facilities and functionalities. This has led to the emergence of new concepts in regard to IoT experimentation.

These include the IoT Testbed-as-a-Service (TaaS) [2] and Experiment-as-a-Service (EaaS) [3].

In [4], a small-scale testbed with cost-effective and easy-to-build design was presented. It was deployed at the building of Electrical Engineering Department of the University of North Texas. The design of the testbed was based on a single base station and a set of sensor nodes running the Zigbee protocol and having different sensors. In addition, a number of software components for sensor data transmission, gateway functionality, database management, and user interface were implemented.

The Twonet testbed was proposed in [5] with a large-scale design. It was deployed throughout a university building of four floors. The main objective of Twonet is supporting multichannel wireless sensor networking. The design of Twonet is based on three-tier architecture. At tier 1, a Linux PC server exists to control testbed access and experimentation through a custom web interface. It also has a set of 20 Raspberry Pi nodes, interconnected using PoE switch, as proxies at tier 2 for collecting debug logs and sensor node data. Each proxy interconnects a set of sensor nodes (at tier 3) implemented using Opal sensor devices.

Another large-scale testbed is SmartCampus [6] located in the Centre for Communication Systems Research (CCSR) building at the University of Surrey, UK. It was developed as part of the European SmartSantander experimental facility [7]. It was deployed across an entire building of three floors in a real-world environment with realistic operational conditions. The testbed had a user-centric design which enables effective study of user behavior in IoT environments and the efficiency of IoT solutions. The testbed was designed following a 3-tier model connecting a number of IoT nodes to different gateways to be then connected to a cloud. Each IoT node was implemented using a TelosB mote with different sensors and a IEEE 802.15.4 radio module. The testbed also incorporates 30 Android Smartphone carried by the users and connected to the gateways. The testbed relies on a management framework for configuring and controlling its functionalities. It provides a user interface enabling resource discovery and reservation, topology display, experiment configuration and execution, and data analysis.

In [8], the testbed was built with a small-scale setup and designed for different IoT application domains. The main objective of the testbed is providing a low-cost, reliable, and customizable testbed. It enables low power consumption, test repeatability, and code portability. The testbed design is based on a number of sensor nodes wirelessly connected a dedicated gateway using nRF24L01+ chipsets with low power consumption. The sensor nodes and gateway were implemented using Arduino UNO and Arduino DUE, respectively. The gateway is connected to a server which runs three software components: Python adapter server, Apache web server, and MySQL database server.

Supersensors were presented in [9] as a campus-wide testbed for the University of Glasgow. It was based on a distributed data collection system with flexible and scalable microcomponent design architecture. The testbed can be used for implementing different use cases for smart campus development. The testbed architecture is composed of a

number of sensor nodes (Raspberry Pi with a set of sensors) connected to a central server. The sensor nodes run different daemons for real-time sensor data collection and network connectivity establishment. At the server side, a publish/subscribe model with a queuing system is implemented for effective real-time data processing and storage in a database. The Protocol Buffers language (proto3) is also used for serializing and parsing data. In addition, a heartbeat mechanism is implemented in the server to enable remote health and availability monitoring as well as reprogramming and configuration of the sensor nodes. The server provides a user interface allowing users to access node and sensor data using REST APIs.

In [10], another IoT testbed deployed at Okayama University of Science across a five-floor academic building was presented. The testbed was constructed using a set of Note PCs and Raspberry Pi devices. A Distributed Internet Traffic Generator software was implemented to generate different types of network data traffic in the testbed. It was also used to obtain information about different metrics in regard to data transmission. The testbed enables effective evaluation of IoT protocols and algorithms considering different realistic networking scenarios and parameters.

In [11], an IoT testbed was developed for dependability competition to quantitatively test low-power wireless sensing systems in environments with high radio interference. The testbed was deployed using a number of TelosB devices with low-power wireless radios in a building at Graz University of Technology in Austria. A Raspberry Pi running a control software is interfaced with each TelosB device using an open-hardware interface board capable of energy consumption monitoring. Users can load their binary images into the boards to run IoT experiments for certain duration. The system provides different performance measures such as delay, consumed energy, and reliability.

An IoT testbed developed following a minimalistic approach to maintain cost-effective implementation was presented in [12]. It was deployed at the Inria Paris across two multistory buildings. The testbed was built using a set of nodes; each has a number of components contained in an OtBox. These include Raspberry Pi, multiple OpenMote devices, and a screen. Each OtBox has a testbed control software running on the Raspberry Pi and connecting to an MQTT broker using WiFi connectivity. Users can use APIs for mote configuration and OtBox management. They also can use the screen to interact with the system using a web-based user interface with a dashboard running as a service on IBM Cloud. The testbed is an open-source and open-hardware system which provides open and remote access.

In [13], the MakeSense testbed which enables real-life and large-scale IoT experimentation for social research was introduced. It enables real-time indoor activities monitoring and situation-aware applications. The testbed design is based on a scalable and flexible client-server model. A set of sensor nodes were deployed as clients and connected to a server using the Lightweight Machine-to-Machine protocol through Broadband Internet. The server is hosted in the cloud for better IoT data analysis and visualization. The testbed provides API access to sensor nodes for remote

management and configuration of available resources. The testbed features different security implementations including the Datagram Transport Layer Security (DTLS), preshared key authentication, and data encryption using Advanced Encryption Standard (AES).

An online testbed for large-scale LoRa experimentations deployed across multiple apartments and buildings was presented in [14]. It has a three-layer design architecture. These are the node, gateway, and cloud server layers. At the node layer, a set of Raspberry Pi were implemented and powered using power banks and solar energy. The gateway node at the next layer is responsible for packet forwarding from the nodes to the server. The server layer provides different functionalities including user management, system control, and job schedule. The testbed provides web-based user interface enabling users to configure the nodes and receive real-time experimentation display.

In [15], an IoT testbed targeting effective combination of IoT technologies and web technologies was presented. It was deployed in a home environment for monitoring and controlling different physical parameters. It has a three-layer architectural design which includes the sensing, middleware, and application layers. A set of sensor nodes were implemented using Arduino devices at the sensing layer. These nodes collect and send IoT data to a gateway PC (at the middleware layer) using the Zigbee protocol. At the application layer, a web server was implemented to enable user access and system configuration using HTTP.

Table 1 summarizes and compares these testbed systems with the proposed one. It can be noticed that most of the presented IoT testbeds target specific IoT communication technologies, sensors, and applications. This work factors in the diversity of IoT technology and the variety in IoT setups and properties. The focus is on facilitating general-purpose and multidimensional support of IoT experimentation while considering different IoT applications. It also envisages the importance of developing an IoT testbed platform for advancing research activities in the College of Computer at Qassim University. The main focus is to build a system that will provide researchers with an efficient IoT facility to promote more effective IoT experimentation.

3. IoT Overview

The IoT technology enables merging the physical and virtual worlds by interconnecting most of the physical objects around us to the Internet. It extends computing capability and network connectivity to everyday objects and enabling them to generate and disseminate data. These objects could include home appliances, wearable devices, medical equipment, and vehicles.

IoT systems are based on the combination of different technological domains such as sensor technologies, networking protocols, embedded systems, cloud integration, data processing, and communication technologies. The IoT technology makes a great push towards digital intelligence and smart control and automation with less human intervention.

The IoT technology opens the horizon for a wide scope of smart applications in different domains such as transporta-

tion, healthcare, agriculture, utilities, and surveillance. Smart city is an example of IoT application that can be applied to realize better management of different aspects such as street light, traffic, and daily services. Another example application is fleet management which enables real-time tracking of fleet assets. In addition, IoT can also be applied for healthcare applications to develop remote patient monitoring and alerting systems.

Most of the IoT applications require large deployment of interconnected IoT nodes as presented in Figure 1. These nodes are usually implemented with small-sized devices limited in computation, memory, and energy resources. Wireless sensor networks are typically established among these nodes to be then connected to the Internet using IP-enabled gateways. Each node is capable of sensing and collecting IoT data targeting specific application. Data communications can be established between the IoT objects and the Internet as well as among the objects themselves.

To this end, IoT relies on different essential hardware and software components. These include sensor devices which are important for reading varying parameters of the physical world. These could be environmental, meteorological, ecological, and medical parameters. Each node also must have a processing unit which can be built using a microcontroller/microprocessor and memory to support data processing and storing. The communication module is also another essential component for any IoT system to enable data transmission to the Internet. It is evident that wireless and low power communication is critically required in this context as most of the IoT nodes would be battery-powered. Example technologies for wireless and low power connectivity are Zigbee, Bluetooth Low Energy (BLE), 6LoWPAN, and LoRaWAN. All these node components need to be powered using a power supply unit which could be a battery or a solar cell.

In IoT systems, sensor nodes collaboratively sense certain parameters and communicate the readings with the gateway which then forwards the data to the Internet. During this, the gateway can also be used to carry out local IoT data preprocessing at the edge of the IoT network. IoT data is typically sent to cloud systems in the Internet for effective data storing and management. IoT applications generate big data that requires further application-specific data processing and analysis.

However, IoT poses different challenges which need to be addressed for different IoT solutions. These include effective support of security and high level of protection against potential IoT attacks. Privacy and the ownership of collected data also become critical concerns, in addition to the need for establishing clear privacy protection policies. Another challenge is concerned with legalization and having a global IoT governance system. It is also challenging to maintain interpretability and support of homogeneity in IoT systems. IoT solutions also require high level of Internet availability with effective cost management particularly for large-scale IoT deployment.

4. System Requirement Analysis

The IoT is an interdisciplinary concept that incorporates variety of computing disciplines such as networking, communication, embedded systems, cloud computing, and data

TABLE 1: Comparison of the proposed testbed with other systems.

Ref.	Scale	Objective	IoT communication	Implemented IoT devices	No of nodes	Sensors	Interface	Cloud integration	Data visualization	Node mobility	Portability
[4]	Small	Support of IoT applications related to environmental monitoring	Zigbee	RPi, Arduino, XBee S2B	6	Temperature, humidity	Web API	Not supported	Not supported	Not supported	Supported
[5]	Large	Support of multichannel WSNs	Dual-radio at 2.4 GHz & 900 MHz	RPi, opal mote, debug board	100	—	Web	Not supported	Not supported	Not supported	Not supported
[6]	Large	Real-life user-centric experimental IoT research facility	IEEE 802.15.4	TelosB mote, sensor board, GuruPlug, Android phones, Android tablets	200	Temperature, light, noise level, motion	Web API	Supported	Supported	Not supported	Not supported
[8]	Small	Support of a customizable and multipurpose experimentation	2.4 GHz radio	Arduino, ATtiny85, ATmega328P, ESP8266, nRF24L01+	—	—	Web	Not supported	Supported	Not supported	Supported
[9]	Small	Support of flexible data collection and processing	WiFi	RPi	10	Temperature, light, sound, motion	Web API	Not supported	Supported	Not supported	Supported
[10]	Small	Support of effective evaluation of IoT protocols and algorithms	Ad hoc	RPi, Note PCs	8	—	API	Not supported	Not supported	Supported	Supported
[11]	Large	Support of the dependability of low-power WSNs in environments with high radio interference	—	RPi, TelosB mote, NavSpark-GL, TI LMP92064	45	Light	Web	Not supported	Supported	Not supported	Supported
[12]	Large	Support of cost-effective large-scale IoT experimentation	IEEE 802.15.4	RPi, OpenMote, display screen	80	Temperature, humidity	Web API	Supported	Supported	Not supported	Not supported
[13]	Large	Support of real-life experimentation for social research on indoor activities	BLE	Arch Pro	100	Temperature, humidity, light, noise level, motion, dust density	Web API	Supported	Supported	Not supported	Supported
[14]	Large	Support of cost-effective LoRa experimentation in a large-scale and realistic environment	LoRaWAN	RPi, LoRa module	50	—	Web	Supported	Not supported	Not supported	Not supported

TABLE 1: Continued.

Ref.	Scale	Objective	IoT communication	Implemented IoT devices	No of nodes	Sensors	Interface	Cloud integration	Data visualization	Node mobility	Portability
[15]	Small	Support of effective combination of IoT technologies and web technologies	Zigbee	Arduino, XBee	—	Temperature, humidity, light, gas	API	Not supported	Not supported	Not supported	Supported
Proposed testbed	Small-large	General-purpose and multidimensional support of IoT experimentation	BLE Zigbee 6LoWPAN LoRaWAN	RPi, Arduino, GrovePi, XBee, 6LoWPAN module, LoRa module, UPS HAT battery	12 (extendable)	A wide range of GrovePi sensors	Web API	Supported	Supported	Supported	Supported

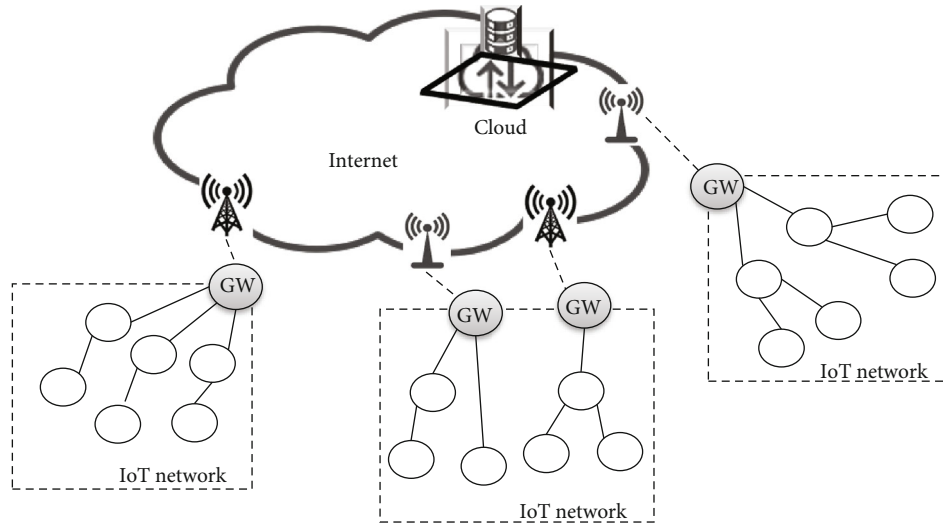


FIGURE 1: Typical IoT deployment.

science. IoT systems have distinct networking requirements in terms of low power communication and lightweight computing. It also relies on easy-to-setup IoT network, cloud integration, and data processing. This makes IoT system challenging to be developed and implemented. For effectively experimenting the different aspects of IoT technology, the design of the IoT testbed should consider different characteristics. These would include efficiency, practicality, simplicity, usability, and interactivity.

IoT architecture consists of different layers as presented in Figure 2. These are the sensing, network, data, and application layers. The testbed should be designed considering this architectural model in a modular design. At the sensing layer, it is important to have a variety of sensor devices that can capture wide range of IoT data. For the network layer, it is required to incorporate the different IoT communication technologies including Zigbee, BLE, 6LoWPAN, and LoRa-WAN. In addition, support of different networking aspects such as data type, network topology, node mobility, network density, and node distribution should be considered. At the data layer, the system should be integrated with a cloud system that enables effective management and processing of the IoT data. The design also needs to facilitate easy use of the testbed to establish certain IoT applications.

Other key requirements include easy control of the system operations. Users should be able to easily configure the testbed for different IoT scenarios and applications. The system should enable the users to have control on which sensors to be used, communication technology to run, and application to build. It is also important to allow the users to configure each setup in terms of different aspects such as sensor reading interval and experiment duration. It is also important to enable the experimentation of different IoT networking problems such as load balancing, failure recovery, QoS, mobility, and security.

This also includes defining the scenario which each node should apply. For example, a setup can be configured to emulate a simple smart home application for 30 minutes using 5 IoT nodes interconnected over a Zigbee network. Each node

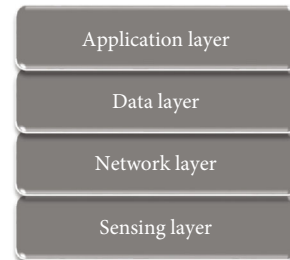


FIGURE 2: The layered IoT architecture.

is configured to read data from a temperature, pressure, light, and motion sensors every 10 seconds and send the data to the cloud. An experimental network scenario can be applied to this setup to have a multihop network topology. During the experiment, node failure can be triggered for certain period in order to examine network performance.

Moreover, simple accessibility of the testbed needs to be realized for the different kinds of potential users. It is also important in this context to allow remote and online access to the testbed. On the other hand, secure design of the system should be realized to at least prevent unauthenticated access and unauthorized use of the system.

There are a number of nonfunctional requirements that need to be fulfilled by the system. These mainly include scalability, portability, flexibility, and cost-effectiveness. The system needs to be easily scaled up at the hardware and software levels. Making the system effectively portable and easily deployable is another important consideration. Different users can have different requirements, thus the design needs to be maintained general and flexible. It is important to keep the development cost to the minimum without compromising its functionality and quality.

5. System Design

The design of the proposed IoT testbed system is based on a modular and scalable architecture, and overview description

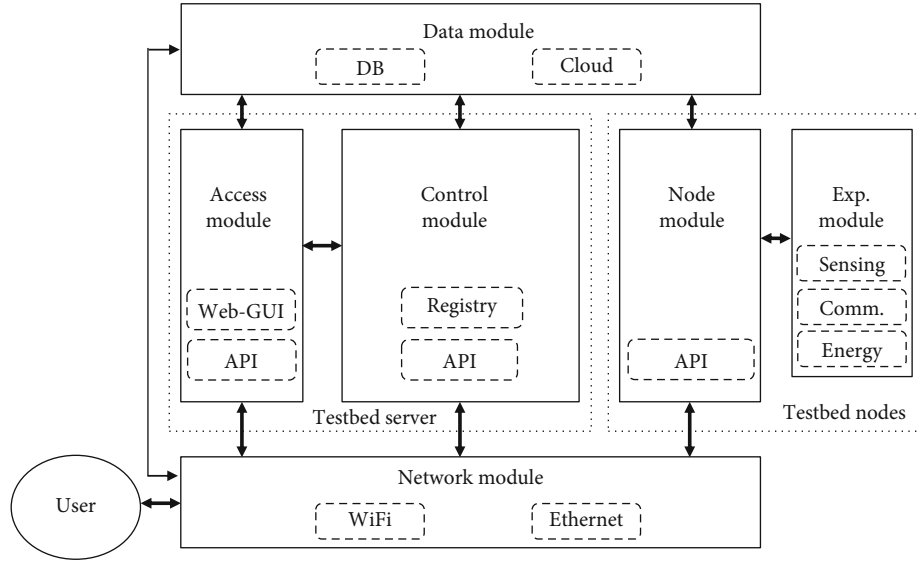


FIGURE 3: Overview of the testbed design architecture.

is provided in Subsection 5.1. A set of hardware and software components were incorporated in the system design as presented in Subsection 5.2. In Subsection 5.3, a detailed operational overview of the testbed functionality is presented.

5.1. Functional Design Architecture. Figure 3 presents an overview of the testbed design architecture. The modular design of the testbed is realized with different functional modules incorporated over the different architectural entities of the testbed. These entities include the central server and testbed IoT nodes. The current design of the testbed is composed of six main modules structured in a systematic manner as follows.

5.1.1. Access Module. The access module is the main front-end layer of the testbed. It facilitates simple and remote access to the system through a web-based Graphical User Interface (Web-GUI). It also enables user access using predefined Application Programming Interface (API) calls. In both cases, the module manages secure access to the system with reliable user authentication and authorization.

5.1.2. Control Module. The control module provides central control and management for the testbed system. It is responsible for monitoring the health and availability of the testbed resources including the IoT nodes. It includes a node registry unit for real-time management of node state. The module is also responsible for managing system functionality and controlling user experiments. It receives user requests from the access module once submitted through the web-GUI. User request can also be received from the user directly. In both cases, the commands are sent as API calls. Then, the control module manages experiment configurations, initiates experiments with the nodes, and controls experimentation setups. Logs on current system state in addition to the data of experiment input and output are collected by the module to be then sent to the data module for permanent storage.

5.1.3. Node Module. The node module manages the different operations of the node entity. It applies requested experiment configurations as received from the control module or directly from the user using the system's APIs. The node module is then responsible for running and controlling experimentation setups. It also communicates real-time data of experiment output with the control and data modules.

5.1.4. Experimentation Module. The experimentation module contains the sensing, communication, and energy units of the node entity. These are the main elements for setting up IoT experiments and applications on the testbed. The experimentation module controls the operations of these units according to the commands received from the node module. Once running an experiment, the module collects sensor readings, communication measures, and energy consumption information. These outputs are then delivered to the node module in a real-time manner.

5.1.5. Data Module. The data module maintains the required functionality for data storage and management. It is responsible of permanently storing system logs, experiment input, and collected experimentation data in the database unit. It also manages user data for enabling secure access to the system. The data module also provides effective integration with web-based cloud systems. The output data of user experiments can be forwarded to the cloud for further data management and processing.

5.1.6. Network Module. The network module provides the underlying communication infrastructure to interconnect most of the system modules. It enables wired and wireless connectivity to the system via WiFi and Ethernet, respectively. The network module is responsible for establishing a system control network among the testbed entities, users, and integrated cloud systems.

5.2. Main Design Components. The testbed design architecture is composed of two main architectural entities of the

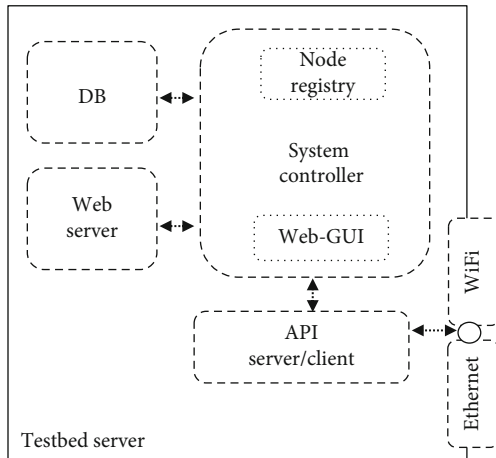


FIGURE 4: The architectural components of the server entity.

testbed. These are the central server and testbed IoT nodes. Both entities are interconnected over a wireless control network providing permanent Internet access.

The server entity consists of a set of functional components as shown in Figure 4. These include the system controller, web server, and database:

- (i) The controller runs the functionality of the control module at the server entity. It facilitates effective testbed management and experimentation control at the main back-end layer of the system. It maintains a node registry service for managing the availability and monitoring the performance of the testbed nodes
- (ii) The web server provides the functionality of the access module at the server entity. It enables web-based access to the testbed system in addition to facilitating easy configuration and setup of the different testbed resources. It also provides the sufficient support to prevent unauthenticated and unauthorized user access
- (iii) The server manages a database that provides local data storage as part of the data module. It mainly retains user access information, system logs, and experiment data

The testbed node entity represents a set of IoT node devices. Each of these nodes has a number of hardware and software components as shown in Figure 5. The hardware components are the sensors, communication devices, and power source:

- (i) The sensors include a set of IoT sensor device that measures a set of parameters such as environmental, biochemical, physiological, behavioral, and visual ones
- (ii) The communication devices enable establishing wireless IoT networks among the IoT nodes. The main IoT communication technologies currently

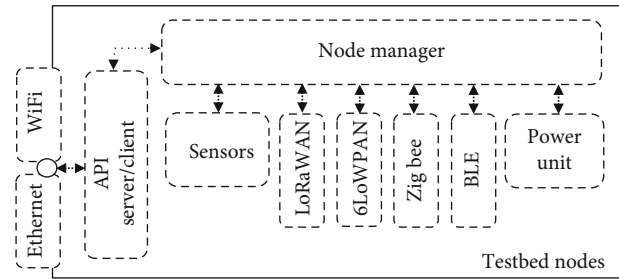


FIGURE 5: The architectural components of the IoT node entity.

supported by the testbed are LoRaWAN, 6LoWPAN, Zigbee, and BLE

- (iii) The design of testbed also relies on portable and sustainable power unit that allows for monitoring power consumption and controlling remaining energy

These components of the IoT nodes support easy reconfiguration, lightweight portability, cost-effective implementation, and effortless deployment. The experimentation module is completely run at the node entity to manage the operations of the hardware components. It enables each component to apply relevant user configurations for effectively running the requested experimental setup.

The node entity also includes a central software component which is the node manager. It runs the functionality of the node module to control the node operations. These operations include initial node registration and frequent status reporting with the system controller. In addition, the node manager controls user experiments according to the requested setup and component configurations. For example, it manages turning on/off specific communication module, initiating sensor reading, and communicating IoT data. The node manager also performs real-time collection and transmission of experiment data to the system controller at the server. The data can also be sent by the node manager to the cloud if required by the user.

For better system usability, the design facilitates easy and simple access via an interactive web-GUI, which enables the following:

- (i) Resource discovery and reservation: the system allows the reservation of a certain number of the nodes, that are currently available, for each experimentation setup
- (ii) Experiment configuration and execution: users can select which communication modules and sensor devices to run for each selected node. There are also entries for setting experiment duration and sensing interval. Users can also provide experimental script to apply certain experiment scenario on the selected resources. This enables setting network topology, setup mode, data traffic type, and packet size
- (iii) Topology display: the interface incorporates an experiment representational display for real-time

visualization of the currently running setup. It visualizes the flow of IoT data among the experimentation resources during each run

- (iv) Data dashboard: the interface provides visual display of collected IoT experiment data in a real-time manner

The system also includes other software components such as the API servers and clients at the server and node entities. User requests and experiment configurations are exchanged among the server and client using predefined API calls. After the initial startup time, each node runs an API server that waits for experimentation calls from the API client. User input via the web-GUI is firstly parsed into API calls and then transmitted by the API client to the API server. These are then processed and forwarded to the node manager for running the requested experimentation setup. The API server also accepts API calls from users directly in the case of having no functioning control module.

The communication of API calls among the different entities is maintained using the network module. This is carried out over a wireless control network established using a private WiFi access point. Using wired Ethernet connectivity is also supported as an alternative option. In both cases, the entities are provided with Internet access.

5.3. Operational Overview of the System. The sequence diagram presented in Figure 6 provides an overview of the main operations of the testbed system. At the startup stage, each node discovers its currently available experimentation modules and hardware components. At the node entity, the node module sends a status check message to the experimentation module which responds with status information of the available experimentation resources. Then, the node module sends a registration request message to the control module. This message contains different node information including its ID, IP address, available experimentation components, and remaining power. The message is processed by the node registry unit to maintain the availability and monitor the performance of the testbed nodes. It then replies to the node module with a response message. The node information is also sent to the data module for permanent storage of the data in the database.

User access to the system is managed by the access module during the login stage. New users are required to firstly register to the system. The registration information is forwarded to the data module for permanent storage. After receiving access request, user data is obtained from the local database to carry out secure user authentication and authorization. Users receive access tokens if and only if the access is legitimate.

User requests for IoT experimentations can be issued through the web interface or API calls. In the former case, experiment configurations and setup script entered by the user are forwarded to the control module in a request message. After copying the request data into the database, the control module forwards the message to the node module using internal API calls. The request is processed by the node

manager software at the testbed nodes selected for the current experimentation. It then replies with a response message which is forwarded by the control module to the display unit for user notification. Upon that, the node manager sends the user configurations to the experimentation module to configure the sensing, communication, and energy components accordingly. After that, a confirmation message is sent back to the node module in order to signal the readiness of the experimentation resources. The node manager can then issue a run command and get the experiment started. It also applies the setup script to control the experimentation scenario accordingly.

For the experiment duration, the experimentation module keeps sending the collected IoT data and measures to the node module in a real-time manner. These are then forwarded to the control module which forwards them to the access module for real-time visualized display on the web interface. The result data is also sent to the data module for local storage and cloud backups.

In the other case, users issue experimentation requests directly to the testbed nodes using API calls. This supports system availability during the failure of the server entity. The request message is received and processed by the node module at each of the selected IoT nodes. The node manager issues a response to the user, copies the request data to the data module for cloud storage, and then sends the user configurations to the experimentation module. It also initiates the experiment after receiving a setup confirmation message from the experimentation module. Once the experiment started, the collected IoT experimentation data is forwarded to the node manager in a real-time manner. The data is then forwarded to the data module to be uploaded to the cloud system for further data processing and visualization. The user then can freely access the data at the cloud anytime the data module using the given cloud API calls.

6. System Implementation

The proposed IoT testbed system was implemented based on integrating a set of hardware components and developing different software modules. A representational overview of the testbed implementation is presented in Figure 7. The following subsections, 6.1 and 6.2, provide a thorough description of the current hardware and software implementations, respectively.

6.1. Hardware Implementation Overview. A collection of open-source and cost-effective hardware components was utilized to develop the current implementation of the testbed. These include a powerful PC laptop which was temporarily set up as the testbed server. It is wirelessly connected to a private WiFi Access Point (AP) which provides access to the Internet and the cloud service. The IoT nodes of the testbed were implemented using multiple Raspberry Pi boards of Version 3 and Model B. The nodes are wirelessly connected to the AP to establish a control network with the server and gain Internet access.

Each Raspberry Pi board runs a stable and open-source Linux-like operating system, namely, Raspbian.

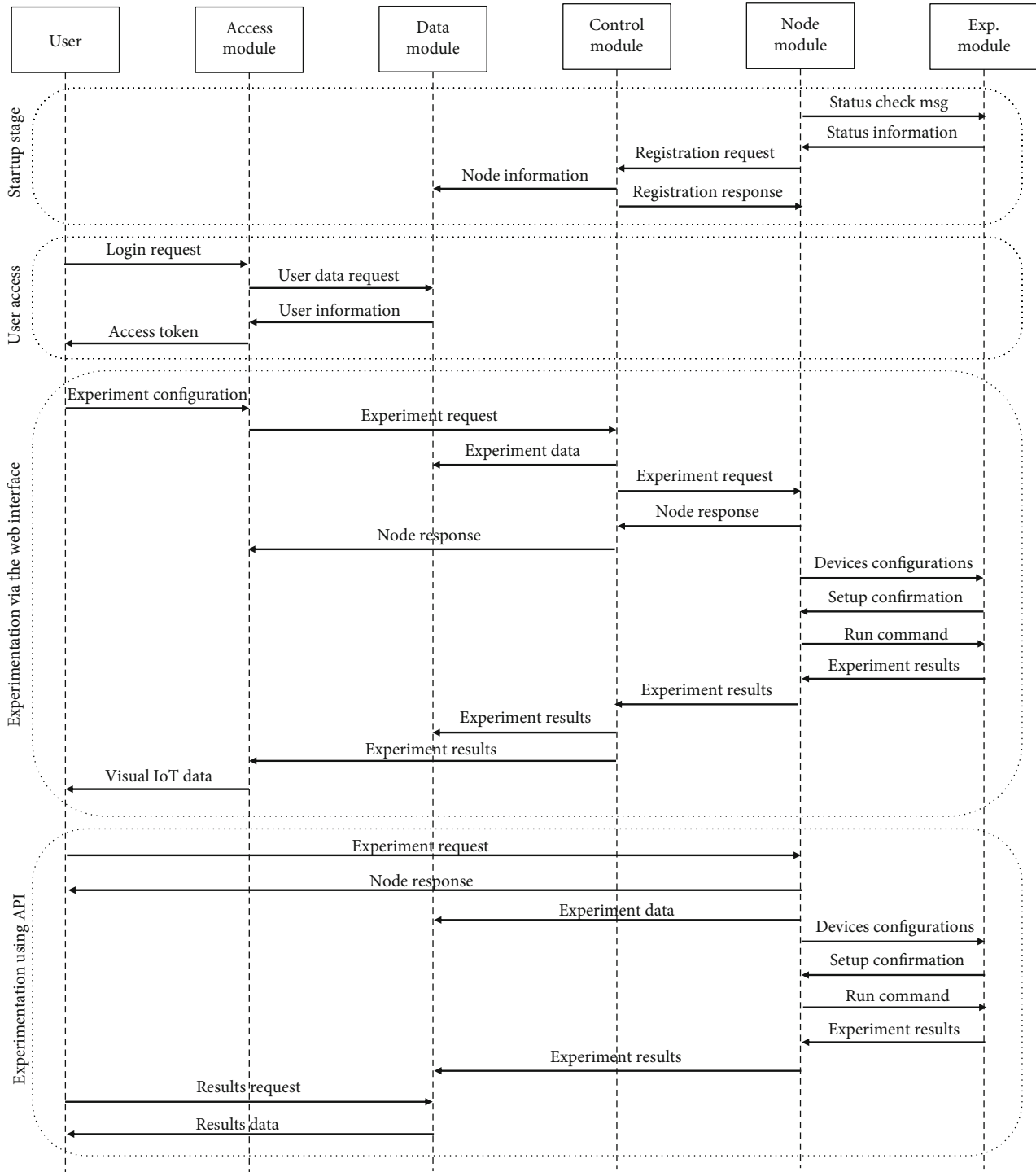


FIGURE 6: Operational overview of the testbed system.

The decision of adopting the Raspberry Pi board was made considering its capabilities in terms of CPU, memory, and connectivity. It also has a built-in WiFi and Bluetooth Low Energy (BLE) modules which are essential components of the testbed. In addition, it supports easy integration of wide range of IoT sensors and communication modules over its I/O interfaces such as the General-Purpose Input/Output (GPIO) pins and serial ports. These were used in the current implementation to attach the fol-

lowing external hardware modules to each Raspberry Pi board:

6.1.1. *XBee S2C*. XBee S2C is an RF module developed by Digi International to support short-range wireless Zigbee communication.

6.1.2. *6LoWPAN Chip*. A tiny 802.15.4 radio chip was developed by OpenLabs to provide support for 6LoWPAN over

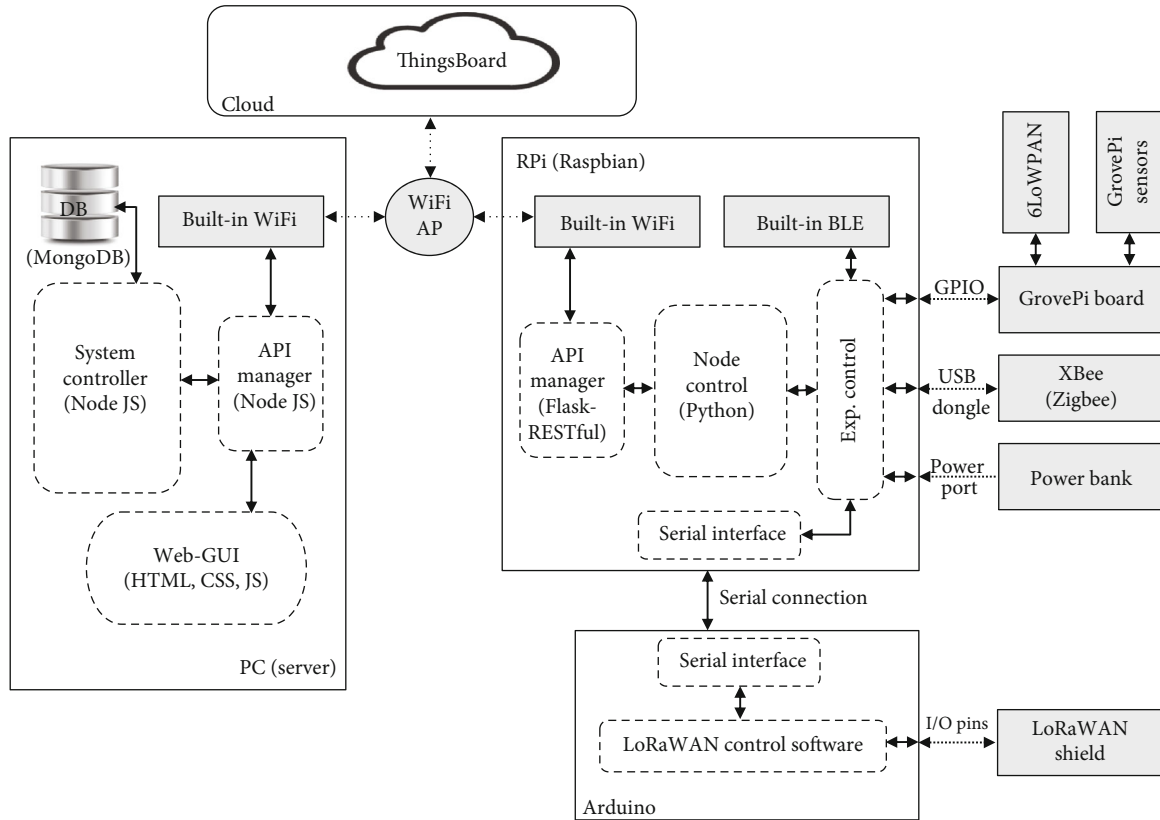


FIGURE 7: Representation of the current implementation of the system.

802.15.4 to the Raspberry Pi board. This RF module is designed to be attached to the GPIO pins of the board.

6.1.3. Arduino Board. Arduino is a microcontroller board that provides an open-source development environment. It has a number of digital and analog I/O pins which can be used for further extensions such as sensor devices and communication modules. There are different versions of Arduino, and the Arduino UNO version is the one being used in the current implementation.

6.1.4. LoRaWAN Arduino Shield. The LoRaWAN shield was developed by Dragino for Arduino UNO boards. It provides a long range transceiver that is based on open-source library. The shield has a set of I/O pins which were used for direct attachment to the Arduino board.

6.1.5. GrovePi+. GrovePi+ is an add-on sensor board developed by Dexter Industries to provide a simple solution for sensor attachment. It provides 7 digital, 3 analog, and 3 I2C ports. The testbed has GrovePi+ boards mounted on each Raspberry Pi board.

6.1.6. Collection of Compatible Sensor Devices. These are connected to the different ports of the GrovePi+ board. These include temperature, humidity, barometer, dust, air quality, gas, light, sound, and motion sensors. However, users are provided with the flexibility to replace these sensors with a variety of alternative sensors to support different IoT applications.

6.1.7. External Power Module. The testbed provides two different powering options for the battery modules. These are a portable power bank and the UPS HAT Battery Adapter Power Supply Extension Board. The testbed also supports the integration of any other power sources such as integrated solar cells.

Each Raspberry Pi board in the current implementation has a GrovePi+ board attached using the GPIO pins. The sensor devices are directly connected to the ports of the GrovePi+ board. The 6LoWPAN module is also attached to the extended GPIO pins on the GrovePi+ board. On the other hand, The XBee S2C module is attached to a customized dongle which is connected to the Raspberry Pi board via its serial port. The communication between the board and the chip is established over a serial connection.

In addition, another serial port is used to connect each Raspberry Pi board to an Arduino board which is used to mount and operate the LoRaWAN module. This can also be used as a further development board to support the scalability and flexibility of the system. However, the Arduino board and LoRaWAN module are completely controlled by the experimentation module implemented in the Raspberry Pi. This is maintained over a permanent serial connection.

In regard to the power module, it can be a power bank that is connected to the micro-USB power input of the Raspberry Pi board. It also could be the Battery Adapter Power Supply Extension Board which can be attached to the GPIO pins.

All the hardware components of each IoT node are contained in a compact-sized and lightweight box in order to support the portability of the system. Users are provided with the ability of placing and moving the IoT nodes in a flexible manner. The current implantation was developed with 12 IoT nodes. However, the scalable design and flexible implementation of the system allows for easily integrating additional IoT nodes in future.

6.2. Software Implementation Overview. A functional implementation of the main modules of the testbed design was developed over the different hardware components of the system. This was accomplished using a set of software development tools as follows:

6.2.1. Access Module. The web-GUI was developed as the front-end of the system using HTML, CSS, and JavaScript.

6.2.2. Data Module. The database was implemented using MongoDB. For the cloud service, the current implementation is integrated with ThingsBoard, a public cloud platform. It provides a set of APIs which was used for managing the communications with the cloud service.

6.2.3. Control Module. The control module's functionalities were implemented as the system back-end using Node JS.

6.2.4. Node Module. The different functionalities of the node module were implemented using Python and Flask-RESTful.

6.2.5. Experimentation Module. The communication, sensor, and battery submodules were also developed using open-source Python classes. As it was attached to an Arduino board, however, the development of the LoRaWAN submodules was carried out in the Arduino C-like environment.

The current implementation of the testbed server includes a web-GUI run by the access module. The interface enables a testbed user to input a setup script that describes the main experimentation setup of an experiment. The script is formed according to the user selections of the number of nodes, communication module, sensors, setup mode, scenario type, and experiment duration. A variety of experiment configurations can also be entered to configure the components of the established setup. These include sensing interval, transmission interval, packet size, battery level threshold, node operational mode, communication channel, and network topology.

The testbed server also incorporates a Node JS implementation of the system controller which provides the different functionalities of the control module. It runs the node registry which relies on a developed heartbeat mechanism for effective remote monitoring of the IoT node health and availability. The mechanism was implemented to frequently examine the server node connectivity to detect any failure.

On the other hand, the system controller was also implemented to receive setup script and experiment configurations in a POST request. The request data is then copied into the experiment table of the system database and forwarded to the selected nodes. The control module handles the communications with the IoT nodes using a set of developed REST

API calls. Received API requests by the node module are handled by the Python implementation of the node control software. It processes the setup script to control the experiment accordingly. The experiment configurations are obtained by the experimentation control module as input to configure its components accordingly. Once the experiment is run, data are obtained and processed by the node control software as JSON data. It is then sent back to the server in response to the API request. The data is frequently obtained by the access module for real-time display on the interface.

7. Use Case

The current implementation of the proposed IoT testbed system enables running effective experiments to examine different aspects of the IoT technology. These include the characteristics of IoT communication in regard to different parameters such as link quality, transmission range, power consumption, and data rate. This can be carried out in experimental scenarios defined according to different considerations including setup size, network topology, hardware configurations, and traffic type. Users can select to run experimental setups using one of the implemented IoT communication technologies, namely, Zigbee, 6LoWPAN, LoRaWAN, and BLE. These would also facilitate studying different properties such as interference and noise for different IoT network setups. In addition, the portability of the testbed nodes enables customized node positioning and allows for flexible establishment of mobility setups and dynamic scenarios. It facilitates studying network performance during the mobility of certain nodes or considering the movement of the whole network under realistic conditions.

Experiment can also be carried out for effective IoT data collection and processing. Since the testbed provides many sensing options using different sensor devices, data can be easily acquired for environmental, meteorological, ecological, and many other purposes. The integration with the cloud systems enables effective processing and simple visualization of the collected data. Therefore, a variety of IoT applications can be established using the testbed for different purposes such as monitoring, control, or automation. Examples include indoor environmental quality measurement, human activity monitoring, smart campus, indoor asset tracking, and smart classroom management.

To present a usage example of the testbed, an experiment was carried out to measure energy consumption of selected IoT communication technologies. In this experiment, the testbed was accessed using API calls which are received and processed by the nodes directly. Output data was then obtained from the node after having the experiment completed. Table 2 lists the main setup and configurations of the experiment. It was carried out over a simple network topology of only five IoT nodes. Three networking setups were developed to interconnect the nodes in a start topology using different IoT communication technologies. These are BLE, Zigbee, and 6LoWPAN. In all the scenarios, however, each node was configured to sense and send temperature sensor data every 5 seconds during a period of 45 minutes. Each setup was repeated for 10 times, and the battery of each node

TABLE 2: Experiment configurations.

Parameter	Setting
Number of nodes	5 nodes
Communication modules	BLE, Zigbee, and 6LoWPAN
Scenario	Data transmission
Duration	60 minutes
Transmission interval	4 seconds
IoT data type	Temperature sensor readings
Data packet size	Low-sized payload
Setup mode	Static
Battery charge status	Fully
Network topology	Star
Number of runs	10 runs

was fully recharged for each run. Data indicating the consumed energy is collected for each run, and the average consumption for each setup is presented in Figure 8. It can be noticed that low energy consumption was maintained in all the setups. However, data transmission over BLE relatively consumed less battery power compared to the Zigbee and 6LoWPAN setups.

The testbed system is also now being used for a new IoT experiment which requires the deployment of the testbed nodes across the building of the College of Computer at Qassim University. The main objective of the experiment is to experimentally examine the effect of IoT node positioning on the overall network performance in indoor environments. The focus is on two different IoT communication technologies which are Zigbee and LoRaWAN. Ten nodes are currently utilized to run different scenarios with different topological setups. In each of these scenarios, the nodes are placed in varying locations inside the building. In some of the scenarios, some of the nodes will be mobile with a simple movement pattern. This would enable to establish different setups considering different criteria such as topological distance and the presence of obstacles.

8. Evaluation

For effective evaluation, the system was assessed using the Technology Acceptance Model (TAM) [16]. It is a commonly adopted method for analyzing technology usage behavior and user attitude. Accordingly, a TAM-based questionnaire was developed for this study in respect to the common TAM factors. The questionnaire was then filled by a total of 46 participants who were involved in the use of the system.

The TAM-based questionnaire was formed with the main TAM factors. These are as follows: perceived usefulness, perceived ease of use, attitude toward usage, and behavioral intention to use. The questionnaire basically measures 16 items according to a 5-point rating scale ranging from “1: strongly disagree” to “5: strongly agree”. Figure 9 presents the TAM model of the system.

The participants were grouped in different groups of 2-3 participants, and some of the participants used the system

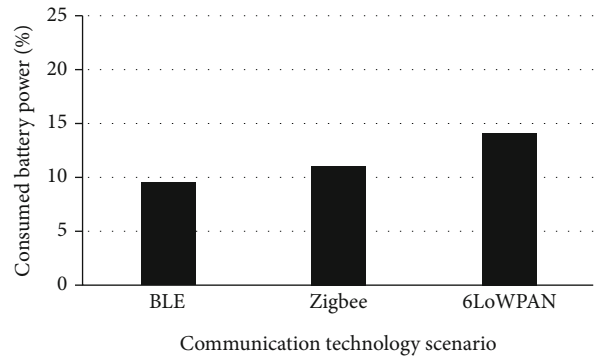


FIGURE 8: Experimental results.

individually. There were a total of 16 different groups and 4 individuals. There was good diversity among the participants in terms of age, educational level, and technical knowledge. That is, age range of the participants was 19-48 years. About 70% of the participants were MSc and BSc students, whereas the rest were PhD and MSc degree holders. In terms of their technical knowledge, most of the participants had no IoT experience, whereas few of them had some IoT knowledge.

The proposed testbed was used by all the participants to test the different functionalities of the testbed. This took place in one of the labs at the College of Computer in Qassim University. The participants were involved in building and running three IoT experiments considering IoT data collection and processing, IoT networking using the different communication technologies, and the development of IoT applications. The main details of these experiments are presented in Table 3. Each experiment was developed for certain user level, ranging from basic to advanced, and configured for specific testing duration. The experiments also varied in terms of the number of nodes, communication technology, and topology setup. The participants also used different types of sensors with varying sensing interval in each experiment. They also experienced different modes to access and configure the system. It is important to note that every participant group and individual was involved in at least two of the experiments; each was conducted for only a single run.

The evaluation was based on the observation of the participants during their use of the testbed. The TAM questionnaire was also filled by every participant individually afterwards. After collecting the evaluation data, the reliability of the TAM results was firstly evaluated. This was based on calculating Cronbach’s alpha coefficient for each of the considered factors. It is a consistency coefficient representing the dimensionality of the TAM questionnaire’s questions. The resulted coefficient values are listed in Table 4. As it can be seen, high value of Cronbach’s alpha coefficient was achieved by each factor. There was no coefficient value lower than the minimum acceptable value of 0.7.

On the other hand, Table 5 shows the evaluation results of the TAM questionnaire. It lists the resulted values of the calculation of the scoring average and standard deviation considering each TAM factor. It can be seen that the proposed IoT testbed was able to achieve good evaluation results

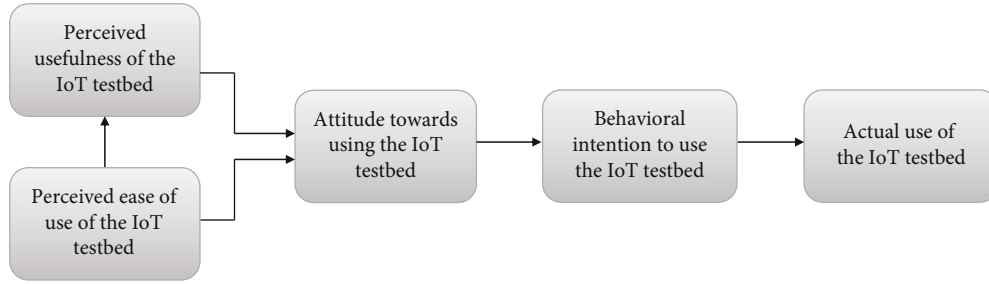


FIGURE 9: The TAM model of the proposed IoT testbed.

TABLE 3: The main details of the experiments for the TAM evaluation.

Setting	Exp. 1	Exp. 2	Exp. 3
Experiment type	Smart home application	IoT data processing	IoT networking
Level	Basic	Intermediate	Advanced
Duration (min)	25	35	45
No. of nodes	6	4	8
Communication tech.	BLE	Zigbee	Zigbee 6LoWPAN LoRaWAN
Network topology	Star	Star	Star Multihop
Sensors	Light Motion Gas	Temperature humidity air quality	Temperature
Sensing interval (sec)	5	10	5
Data packet size	Small	Small	Small-large
Access mode	Web-GUI	Web-GUI API	API
Data storage	Cloud	Cloud Local	Local
Participant no.	13 groups 3 individuals	15 groups 4 individuals	8 groups 1 individual

TABLE 4: Cronbach’s alpha coefficient results.

Factor	Cronbach’s alpha coefficient
Perceived usefulness	0.914
Perceived ease of use	0.882
Attitude toward usage	0.931
Behavioral intention to use	0.844

TABLE 5: Average scoring results.

Factor	Average	STD
Perceived usefulness	4.41	0.71
Perceived ease of use	4.34	0.76
Attitude toward usage	4.25	0.68
Behavioral intention to use	4.23	0.79

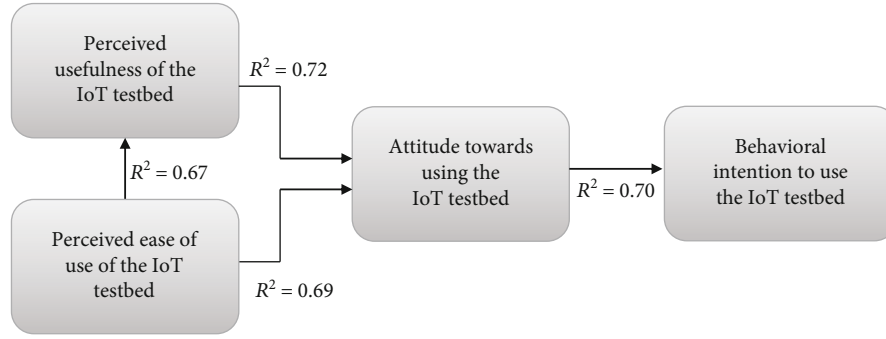
as high average scores and low standard deviation was obtained for all the factors. It is evident that the testbed succeeded in maintaining usefulness and simplicity when it comes to realistic IoT experimentation. In addition, the testbed provides an attracting and motivating experimental IoT facility for the different users as they showed a positive attitude toward the testbed with good intention to use it in future for IoT experimentation.

Furthermore, the correlation between the considered TAM factors was measured and analyzed based on the calculation of the Pearson correlation (PC). This is a critical statistical indicator for analyzing how the dependency relationship

formed among the factors. The calculation results of the Pearson correlation are listed in Table 6. It can be noticed that there was a positive correlation among the considered TAM factors for the proposed IoT testbed. The perceived usefulness factor well-correlated with the ease of use factor. Both of these factors had good correlation with the attitude toward the usage of the system. It is accordingly evident that the testbed attracted the different users to utilize its functionality as a result of its simplicity and efficiency for effective IoT experimentation. In addition, the relationship among the attitude toward usage and intention to use factors achieved a higher correlation value. This clearly indicates that the

TABLE 6: Relationships among TAM factors.

Independent factor	Dependent factor	PC	R^2
Perceived ease of use	Perceived usefulness	0.798	0.67
Perceived usefulness	Attitude toward usage	0.817	0.72
Perceived ease of use	Attitude toward usage	0.791	0.69
Attitude toward usage	Behav. intention to use	0.836	0.70

FIGURE 10: The TAM model with the R -square values.

system was able to maintain high level of user satisfaction and attract user interest for future IoT experimentation.

To determine the variance of the independent factors, the R -square is also calculated and listed in Table 6. It indicates the closeness of the data to the fitted regression curve. Figure 10 shows the calculated R -square values on the TAM model of the proposed IoT testbed. It is evident that these results statistically present effective variation in the response variable.

9. Conclusion

An effective IoT testbed system that takes IoT experimentation to a further level of simplicity and practicality was presented in this paper. The system provides a motivating experimental testbed environment for effective IoT experimentation. It enables researchers to carry out IoT experiments targeting the different aspects and applications of the IoT technology. The testbed design provides a usable, scalable, portable, and flexible system. Testing and experimenting IoT solutions was made inspiring in simple and practical design architecture. The current implementation of the testbed system was accomplished using a collection of efficient and cost-effective components. The results of the TAM-based evaluation demonstrated how the proposed testbed system is usable and useful for conducting experimental IoT studies in a realistic manner. The testbed received positive attitude in regard to its efficiency for IoT experimentation. The testbed was also found to be effective for future use in future IoT research works.

Given its design modularity and flexibility, the testbed can be extended to target more customized IoT experiments. In a future work, the testbed will be utilized to carry out a number of IoT experiments to study IoT network performance considering dynamic and mobile network setups. It is also envisaged that the system can be deployed for a smart

classroom application which would require integrating and comparing some artificial intelligence techniques.

Data Availability

According to the funding policy of this work, data cannot be shared or made publicly available during the funding contract.

Conflicts of Interest

The author declares that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the Deanship of Scientific Research, Qassim University, according to the agreement of the funded project No. SRD-coc-2018-1-14-S-5142. The author thanks the sponsor of this work for their support.

References

- [1] G. Z. Papadopoulos, A. Gallais, G. Schreiner, E. Jou, and T. Noel, "Thorough IoT testbed characterization: from proof-of-concept to repeatable experimentations," *Computer Networks*, vol. 119, pp. 86–101, 2017.
- [2] M. Hossain, S. Noor, Y. Karim, and R. Hasan, "IoTbed: a generic architecture for testbed as a service for Internet of Things-based systems," in *2017 IEEE International Congress on Internet of Things (ICIOT)*, Honolulu, HI, USA, June 2017.
- [3] T. W. Edgar and T. R. Rice, "Experiment as a service," in *2017 IEEE International Symposium on Technologies for Homeland Security (HST)*, Waltham, MA, USA, April 2017.
- [4] S. Ferdoush and X. Li, "Wireless sensor network system design using Raspberry Pi and Arduino for environmental

- monitoring applications,” *Procedia Computer Science*, vol. 34, pp. 103–110, 2014.
- [5] Q. Li, D. Han, O. Gnawali, P. Sommer, and B. Kusy, “Twonet: large-scale wireless sensor network testbed with dual-radio nodes,” in *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*, Roma, Italy, November 2013.
- [6] M. Nati, A. Gluhak, H. Abangar, and W. Headley, “Smart Campus: a user-centric testbed for Internet of Things experimentation,” in *2013 16th International Symposium on Wireless Personal Multimedia Communications (WPMC)*, Atlantic City, NJ, USA, June 2013.
- [7] L. Sánchez, J. A. Galache, V. Gutiérrez et al., “SmartSantander: experimentation and service provision in the smart city,” in *Proceedings of the Future Network & Mobile Summit*, Warsaw, Poland, June 2011.
- [8] A. Di Nisio, T. Di Noia, C. G. C. Carducci, and M. Spadavecchia, “Design of a low cost multipurpose wireless sensor network,” in *2015 IEEE International Workshop on Measurements & Networking (M&N)*, Coimbra, Portugal, October 2015.
- [9] K. Hentschel, D. Jacob, J. Singer, and M. Chalmers, “Supersensors: Raspberry Pi devices for smart campus infrastructure,” in *2016 IEEE 4th International Conference on Future Internet of Things and Cloud (FiCloud)*, Vienna, Austria, August 2016.
- [10] H. Oda, E. Kulla, R. Ozaki, and N. Nishihara, “Design of an adhoc testbed for IoT and WSA applications using Raspberry Pi,” in *Proceedings of the 11th International Conference on Broad-band Wireless Computing, Communication and Applications (BWCCA)*, Asan, Korea, November 2016.
- [11] M. Schuß, C. A. Boano, M. Weber, and K. Römer, “A competition to push the dependability of low-power wireless protocols to the edge,” in *Proceedings of the 14th International Conference on Embedded Wireless Systems and Networks (EWSN)*, Uppsala, Sweden, February 2017.
- [12] J. Munoz, F. Rincon, T. Chang et al., “OpenTestBed: poor man’s IoT testbed,” in *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, Paris, France, France, April-May 2019.
- [13] J. Jiang, R. Pozza, N. Gilbert, and K. Moessner, “MakeSense: an IoT testbed for social research of indoor activities,” 2019, <http://arxiv.org/abs/1908.03380>.
- [14] W. Wang, X. Guo, L. Liu et al., “Dandelion: design of an online large scale LoRa testbed,” in *Proceedings of the International Conference on Embedded Wireless Systems and Networks (EWSN)*, Beijing, China, February 2019.
- [15] A. A. Reshi, A. Alsaeedi, and S. Shafi, “Development and web performance evaluation of Internet of Things testbed,” in *2019 International Conference on Computer and Information Sciences (ICCIS)*, Sakaka, Saudi Arabia, April 2019.
- [16] F. D. Davis, “Perceived usefulness, perceived ease of use, and user acceptance of information technology,” *MIS Quarterly*, vol. 13, no. 3, pp. 319–340, 1989.