An IoT-based Coastal Recreational Suitability System using Effective Messaging Protocol

Farashazillah Yahya[1]*, Ahmad Farhan Ahmad Zaki[2], Ervin Gubin Moung[3]
Hasimi Sallehudin[4], Nur Azaliah Abu Bakar[5], Rio Guntur Utomo[6]
Faculty of Computing and Informatics, Universiti Malaysia Sabah, Kota Kinabalu, Sabah, Malaysia[1, 2, 3]
Faculty of Technology and Science Technology, Universiti Kebangsaan Malaysia, Selangor, Malaysia[4]
Razak Faculty of Technology and Informatics, Universiti Teknologi Malaysia, Kuala Lumpur, Malaysia[5]
[6]School of Computing, Telkom University, Indonesia[6]

***Abstract*—Coastal recreational activities are one of the main attractions for local public beachgoers and overseas tourists. The accessibility to better-quality coastal water will enhance safety and public health awareness when the information is available. Existing platforms showing the risk of whether a beach is suitable for public recreational use is less available in Malaysia. The Internet of Things (IoT) based system design specifically for coastal recreational suitability may differ from the existing configuration depending on the environment and requirements. This paper reports the design and implementation of an IoT-based system to capture the coastal environmental data and recommend recreational suitability. The system captures sensor data, store it in a database and displays the result using a dashboard. The variable data include the temperature, humidity, rain, pH, turbidity, oxidation-reduction potential (ORP), and total dissolved solids (TDS) in a coastal area. The hardware used in the design is the development boards such as Raspberry Pi, Arduino Uno, and ESP32 controller. The system is developed using PHP, MySQL, and Apache Web Server and can be accessed online at https://ipantai.xyz. When using Message Queuing Telemetry Transport (MQTT) as the effective messaging protocol and HiveMQ broker, the result has shown improvement for message size, throughput, and power consumption. The further potential of an IoT-based system is to bring value for coastal management and serve as a powerful tool to determine whether the coastal area is suitable for the public to access water recreational activities.**

Keywords*—Coastal recreational; internet of things; message queuing telemetry transport; sensor; water quality

I. INTRODUCTION

Most countries in the world are surrounded by the ocean. The length of coastlines in the world is estimated at 1.16 million kilometres. Malaysia has about 4,800 kilometres of coastline comprising two distinctly different physical formations, namely the mangrove-fringed mudflats and sandy beaches. The coastlines are about equally divided between mud coast and sandy beaches. At least 50% of the adult population in the world visits the public water areas such as beaches, islands, etc. for recreational purposes [1]. Beachgoers rarely receive water quality information, whether the waters are suitable for recreational activity or vice versa. Poor water quality (such as microbial contamination) in coastal water poses a public health threat due to waterborne diseases. The risk of exposure to microbial contamination can be reduced by informing beachgoers in advance about the water quality. Recreational water activities improve when the water has better quality [2], [3] but users do not always recognise poor water quality or its associated health risks [1]. Coastal recreational activities are rapid development of economy and urbanisation can promise highly to expose to nearest water resources including the coastal area that ultimately embraces poor water quality severely by time excluding the existing harmful microbiology in the coastal ocean area. Therefore, understanding the problems and trends of water quality are crucial and significant to determine the water quality at a specific coastal area, formulate prevention and control, also to educate the public in terms of health risk by establishing related public information and regulation at the coastal recreational area.

In the 21st century, the accessibility to data and information is known at the fingertip. The Internet of Things (IoT) has become the basis of digital transformation and automation in delivering new ideas and service offerings to improve the way we live, work, and entertain ourselves [4], [5]. IoT is foreseen as a technology enabler in various cases. This paper reports the design and development of an IoT-based system for coastal recreational suitability. The system serves as a means to capture and monitoring for beach water quality. The system can be accessed by the public and potentially provide an initial awareness of clean coastlines. The coastal recreational area effectively contributes not just to public attraction as part of a tourism attraction but also contributes to the comprehensive economic driver for the state [6]. To understand and propose a solution to overcome the issue, this paper intends to explore an integrated approach suitable for coastal recreational suitability. The objectives are to:

- Design and develop an IoT-based system for coastal recreational suitability systems.
- Implement an IoT-based system to suit the requirements for capturing, storing, and reporting coastal environment data.

The paper is organised into several sections. The first section introduces the research idea. The next section presents the related work in the area. Followed by methodology for system development and experiment set up. Afterwards, the

*Corresponding Author

paper discusses the results and finally summarise them with a conclusion and recommendation for future work.

II. Related Work

A. Internet of Things (IoT)

IoT can be described as "an interconnection of machines and devices through the internet allowing the creation of data that can shed light on analytic performance and support new technologies” [7]. Another author [8] describe IoT as "a system of a physical object having independent communication among them." Researcher [9] defines IoT as "inter-net-connected embedded systems that can be upgraded and adapted to changing needs on-demand, useful information can be immediately collected from remote geographic areas, and fault diagnosis and system restart can be made more efficient and cost-effective by not having to send out technicians to remote places". Throughout the years, IoT is maturing and continues as one of the exciting concepts in the IT world [10]. Over the last decade, the term IoT has attracted attention by projecting the vision of a global infrastructure of networked physical objects, enabling any time and place connectivity for anything and not just for anyone through the internet [11], [12].

B. IoT in Coastal Research

Some research has been undertaken on coastal recreation using IoT. Be Right Beach (BRB) was developed consisting of a sensor network with a UV sensor, thermometer, humidity sensor and a camera for crowdedness estimation [13]. The data was collected by a cloud platform that provides useful information about the beaches and suggestions where to go based on user preferences like weather, crowdedness, time of travel etc. Another research example was blockchain innovation and IoT will help all users to get involved together in economic activities [14]. Next, researchers found out that the design, development, and deployment of an IoT-based marine environment monitoring and protection system is needed to address some critical issues including autonomy, adaptability, scalability, simplicity, and self-healing [15]. Another researcher [16] suggests application layer is to provide smart application services to meet user needs. In IoT-based marine environments, the application layer covers water quality monitoring, coral reef monitoring, marine (either offshore or deep-sea), fish farm monitoring and wave.

C. Water Quality

The context of water quality has been recognised to be the main highlight that requires important information about public health, environmental concern, and the quality of beach water for public usage. Over the years, coastal water was habitually exposed to anthropogenic and industrial pollution that led to negative consequences to public health and recreational activities [17]. In general, water quality is described as the assessment of sanitary and microbial water quality collectively. The results of the assessment should give a proper clarification about the water quality around the investigated beaches to inform the public, provide on-site guidance and information to the public relative to the safety aspect, assist and promote effective recreational management and formulate regulatory compliance based on the recreational area. Fig. 1 shows the classification matrix of water quality assessment.

D. Recreational Suitability

The detection of faecal contamination at the coastal beaches is subjected to extensive water quality that includes wastewater discharge, industrial waste over and surface runoff. As a result, the suitability of recreational areas is as crucial as the conservation of recreational areas by providing informative and presentable data about water quality trends in the coastal area. One of the recreational suitability assessments is from the Suitability for Recreational Grade (SFRG) [17]. The grading for the recreational suitability is determined by SFRG via the relationship between Microbiological Assessment Category (MAC) and Sanitary Inspection Category (SIC) [17]. Based on the recreational suitability assessment from SFRG via the relationship between the MAC and SIC, proper recreational management will be implemented to measure proper risk management action and formulate legal regulation before the public safety at the recreational area. There are four major field interventions which are compliance and enforcement control and abatement technology, public awareness and information and public health advice and interventions [17]. In this research outlining a better water quality assessment system is crucial to give better guidelines to formulate legal regulation and action plans based on the data generated by the water quality system used at the recreational area.

There has been less research conducted in IoT for coastal recreational suitability. Previous research focuses on environmental water quality using IoT. Similar research applies IoT devices to monitor beaches and crowd detection [13]. Several research reviews on [18] LoRaWAN implementation as an effective for IoT-based monitoring systems with a frequency below 1GHz which and some using GSM and Thingspeak [19]. In this research, we look at the IoT protocols specifically the IoT design for coastal recreational suitability using Message Queueing Telemetry Transport (MQTT) protocol.

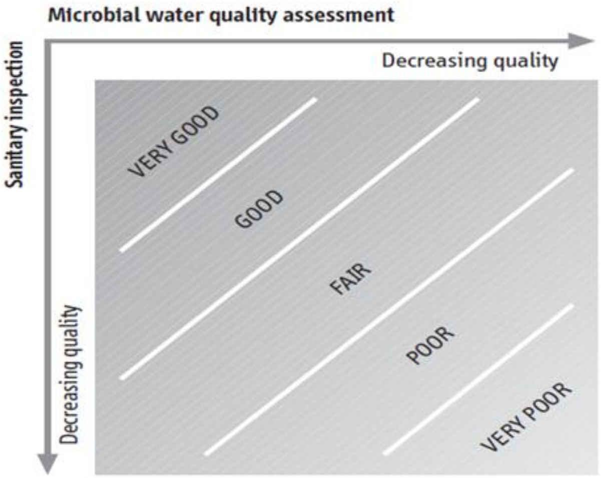


Fig. 1. Classification Matrix of Water Quality Assessment [17].

III. Methodology

A. System Development

In this project, the prototyping method is selected as the approach for system development. A prototype is a software model and through the prototype interaction, users can understand the desired system better. The prototyping method works best in situations where the requirement of the project is unknown. In addition to that, it is a method of iterative, trial, and error that takes place between the system developer and the client.

The advantages of using prototype methodology are:

- It will be easier for the clients to understand the design of the project.
- Real-time monitoring and continuous client-developer interaction assist to build a good relationship.
- Risk factors are easily identified along with steps to mitigate the risk factor.
- Developer can get any input from a client earlier in the development cycle and save time.

In this research, the steps involved in prototyping methodology in Fig. 2 are described.

1) Requirement analysis and information gathering: The first step of prototype methodology involves understanding the very basic requirements of the product, particularly about the user interface. It is possible at this point to neglect the problem in IoT and the messaging protocol that has been developed.

2) System design: The next step is the design the system. As this project must create a web-based system as it is included in client requirements, there need to use an HTTP protocol that relates to a database and world wide web (WWW). To integrate the MQTT protocol in a web-based system, Socket is needed and there is also a need for a Platform as a Service (PaaS) in the computing model. Heroku is used because this platform equips us with a ready runtime environment and application servers. Heroku will be designed as a medium (server) to create a connection to the MQTT broker so it can subscribe to the message from the broker itself. Every browser will request an HTTP upgrade so they can receive the real-time data using SocketIO. The sensor is hardware that is being chosen to measure certain parameters such as humidity, temperature, turbidity, and pH.

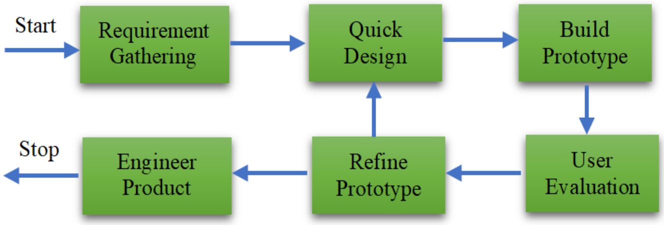


Fig. 2. Prototyping Model.

- HiveMQ

The research proposes the cloud HiveMQ as a broker. The reason for using the cloud was because it is cost-effective and lightweight. The setup for coastal recreational involves many locations, a centralised broker is needed in Fig. 3, and a cloud broker is appropriate. HiveMQ cost is free for a certain limit and suitable for a prototype project. The sensor will publish the data to the HiveMQ broker, and it will publish the message to the client who subscribes to it. Since MQTT are bidirectional communication, it can subscribe and publish at the same time.

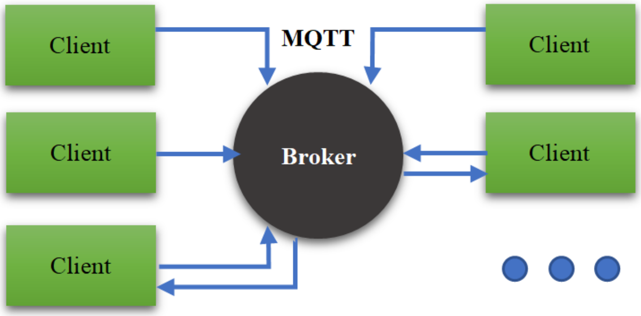


Fig. 3. MQTT Architecture.

- Socket.IO

Socket.IO is a JavaScript library for real-time web applications. It enables real-time, bi-directional communication between web clients and servers. It has two parts: a client-side library that runs in the browser, and a server-side library for Node.js. Both components have a nearly identical API.

Since the project needs to communicate with a broker, socket io can communicate with the MQTT broker and is suitable for this project. Socket.IO primarily uses the WebSocket protocol with polling as a fallback option while providing the same interface. Although it can be used as simply a wrapper for WebSocket, it provides many more features, including broadcasting to multiple sockets, storing data associated with each client, and asynchronous I/O. Socket.IO provides the ability to implement real-time analytics, binary streaming, instant messaging, and document collaboration.

- Apache (HTTP)

Since MQTT does not have a database feature, using the HTTP protocol, it becomes possible to store the data. For the first process of HTTP, the sensor collects the data and transfers it to the Raspberry Pi. The Raspberry Pi, which acts as a web server and uses as a local host for the initial prototype. The sensor is connected to Arduino and Raspberry Pi. The port is going to use port 80. Fig. 4 shows how the HTTP protocol works initially started when a client requested a packet. Send the HTTP header using the TCP/IP protocol and the webserver is decoding, creating the header and formatting data before sending the HTTP response header back to the client.

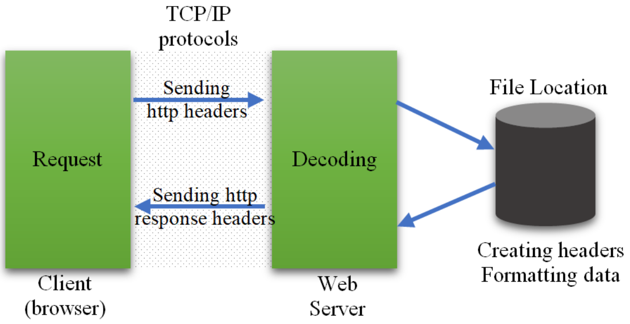


Fig. 4. HTTP Architecture.

3) Developing initial prototype: In this step, the initial prototype is created, demonstrating the basic requirements, and providing user interfaces. It will implement the MQTT protocol with HiveMQ as a cloud broker. Then it will be connected to the Socket IO server that will subscribe to HiveMQ data. Once it gets the data, it will broadcast the message to the client whenever connected to the server. The initial prototype has an explore module that will retrieve the MQTT data in real-time on a webpage. This initial prototype will be presented to the client for client evaluation.

4) Client evaluation: The next step is the client initial evaluation. In this step, the proposed system will be presented to the client for an initial evaluation. The user will verify the function created and it will tally with the client requirement itself. It will help to determine the strength and weaknesses of the coastal recreational system. Comments and suggestions will be collected from the client and provided to the developer.

5) Refining prototype: In this phase, if the client is not satisfied with the current prototype, the prototype must be improved according to feedback and suggestions from the client. For example, every time was adding new function and module, the user will verify and check the requirement. For example, in this system development, the first module been created are explore module, then the home page module, login/signup module, sensor entry module and sensor reading module. This phase will not be completed until all the client specified requirements have been met. Once the client is pleased with the existing prototype, based on the approved final prototype, a final system is produced.

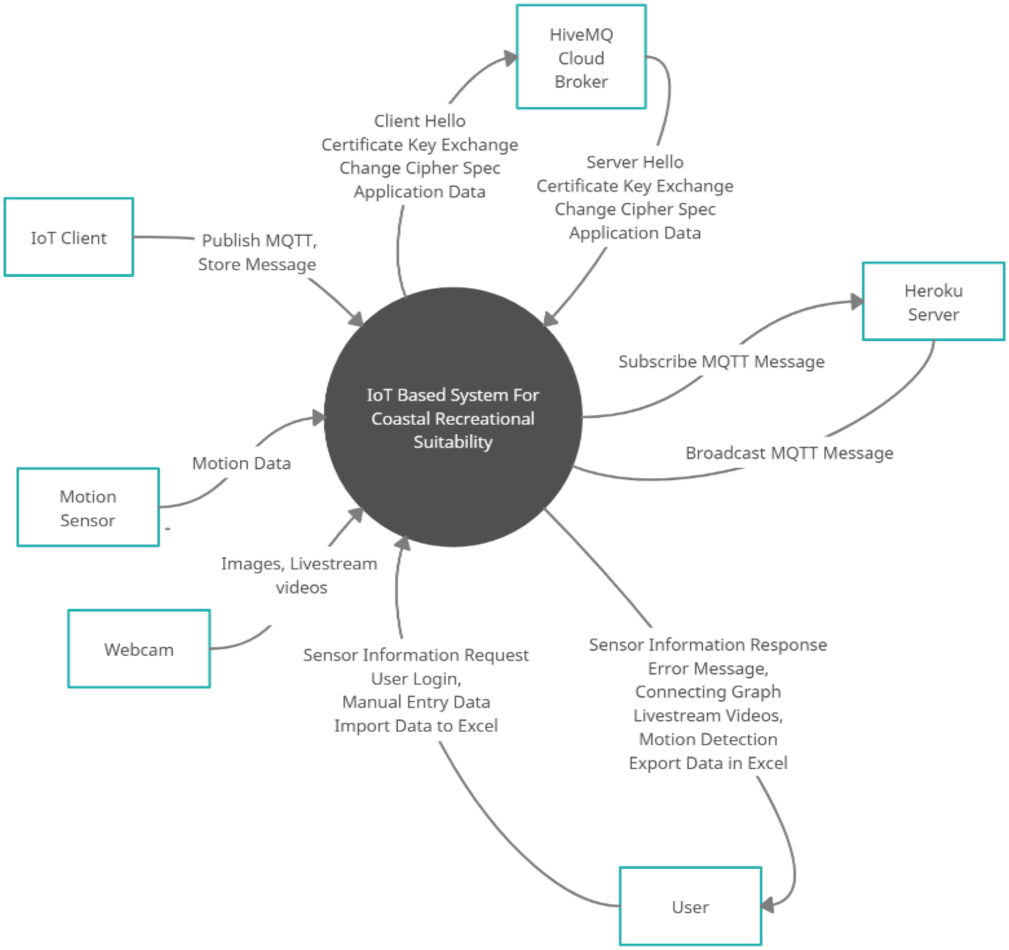


Fig. 5. Context Diagram IoT based Coastal Recreational Suitability System.

6) Implementation and maintenance: Once the final system is developed based on the final prototype, it is thoroughly tested and deployed to production. The system undergoes routine maintenance to minimising downtime and prevent large-scale failures. For example, there is a need to check the sensor onside rather than remote access periodically and to make sure the sensor is cleaned and no obstructions. The system was also tested on a web-based system by checking on Cloudflare to see if there are any attacks on the webserver.

B. Network Design

In this project, three locations are being presented. The system displays based on single locations. For example, the ODEC beach is shown in Fig. 6. These single locations are connected to the HiveMQ cloud broker to publish the message. Socket IO is implemented, and it subscribes to the messages from the HiveMQ broker. Furthermore, SocketIO will publish a broadcast message to every connected client that requests to view the data. Clients that view the data must request io.emit that has been applied on server ipantai via JavaScript. The JavaScript will automatically emit socket io to the server to make a TCP connection via SocketIO. If the connection is successful, every message transmitted by the sensor will be broadcast by the socket io server to the client.

IV. System Implementation

System analysis is gathering and interpreting data to recognise the problems and decomposition into its component. In this phase, the requirements and functionality of this project are determined. The technique used to conduct this analysis is to compare the benchmark with another similar project that uses MQTT with different parameters. Parameter involved for comparison is the message size of MQTT, throughput, and power consumption. After the process of data gathering and comparison, a prototype is developed. Some features in the existing system are based on interviews with stakeholders which are new and unique. Fig. 5 shows the context diagram of the IoT Based System for Coastal Recreational Suitability with six entities which are IoT Client, HiveMQ Cloud Broker, Heroku Server, Motion Sensor, Webcam and User.

Fig. 7 shows a use case diagram of this system works as there are 16 uses cases for this system. For the MQTT part, only five uses case is involved while the remaining 11 are from client requirements. In this use case, three core parts will have their function such as HiveMQ Broker, Heroku Server and ipantai Webserver.

Fig. 8, 9, 10 and 11 show the user interface for the Coastal Recreational Suitability System. The system can be accessed at https://ipantai.xyz.

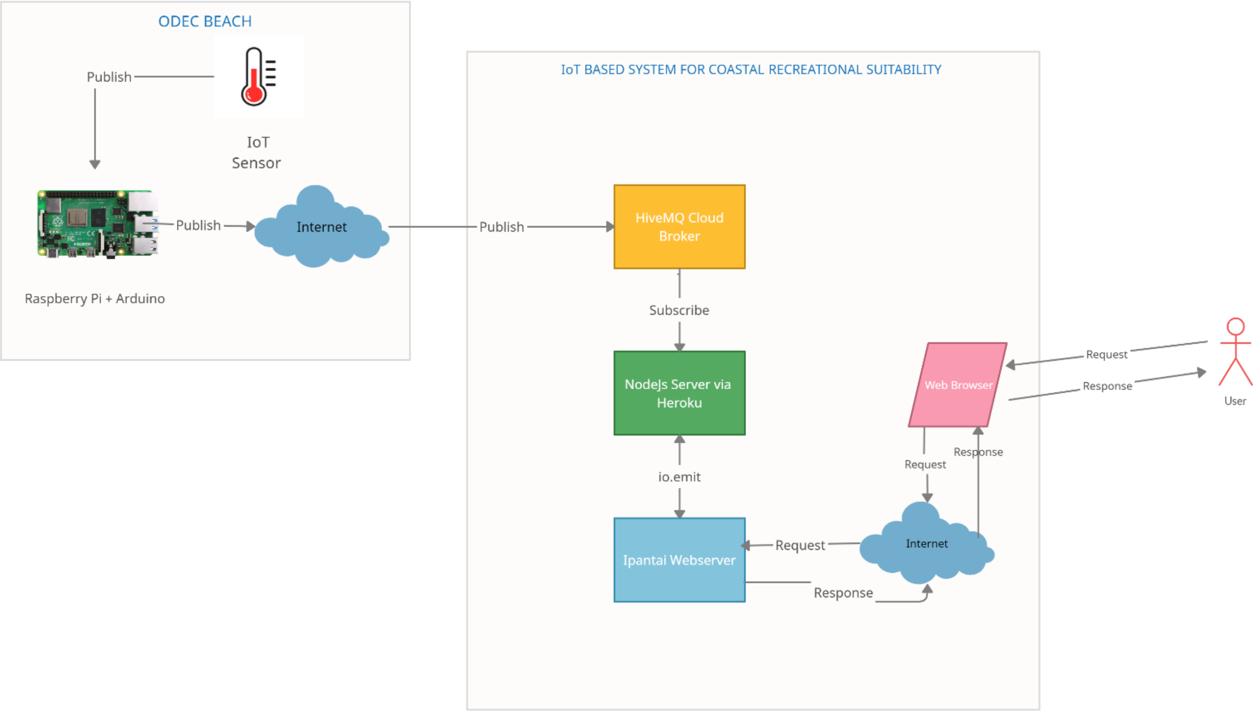


Fig. 6. The Network Design.

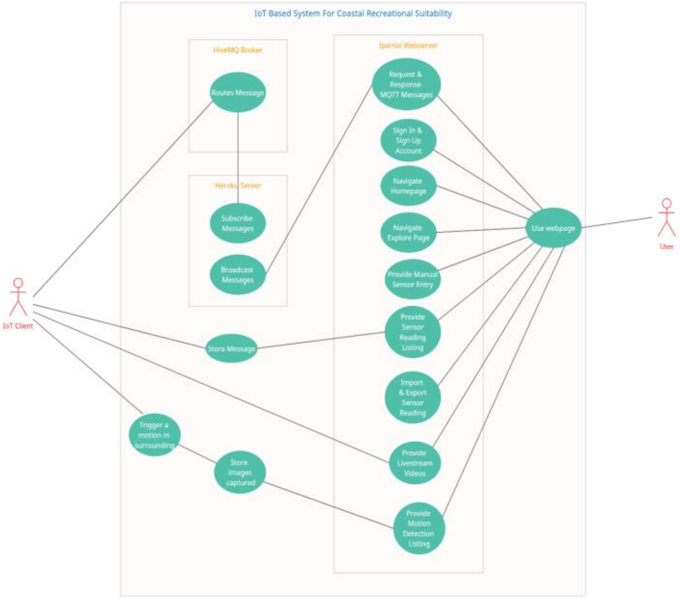
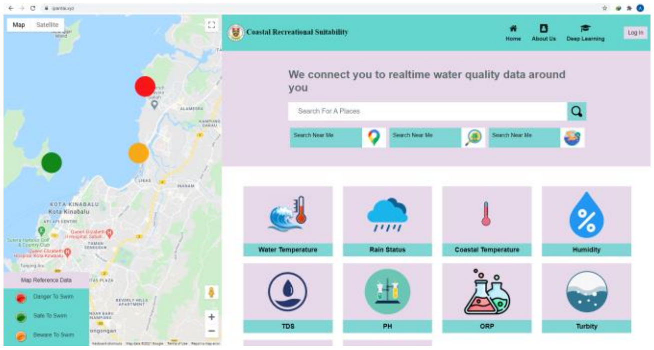
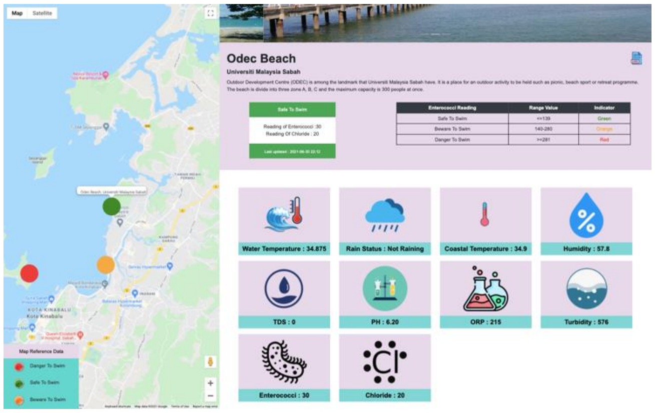

IoT Based System For Coastal Recreational Suitability
HiveMQ Broker
Routes Message
Heroku Server
Subscribe Messages
Broadcast Messages
Ipantai Webserver
Request & Response MQTT Messages
Sign In & Sign Up Account
Navigate Homepage
Navigate Explore Page
Provide Manual Sensor Entry
Provide Sensor Reading Listing
Import & Export Sensor Reading
Provide Livestream Videos
Provide Motion Detection Listing
Use webpage
User
IoT Client
Store Message
Trigger a motion in surrounding
Store images captured

Fig. 7. The use Case.



We connect you to realtime water quality data around you

Fig. 8. The Main Page of the Coastal Recreational Suitability System.



Odec Beach

Universiti Malaysia Sabah

Outdoor Development Centre (ODEC) is among the landmark that Universiti Malaysia Sabah have. It is a place for an outdoor activity to be held such as picnic, beach sport or retreat programme. The beach is divide into three zone A, B, C and the maximum capacity is 300 people at once.

Reading of Enterococci :30
Reading Of Chloride : 20

Enterococci Reading	Range Value	Indicator
Safe To Swim	<=139	Green
Beware To Swim	140-280	Orange
Danger To Swim	>=281	Red

Fig. 9. The Page of Specific Beaches showing the Variable Data.

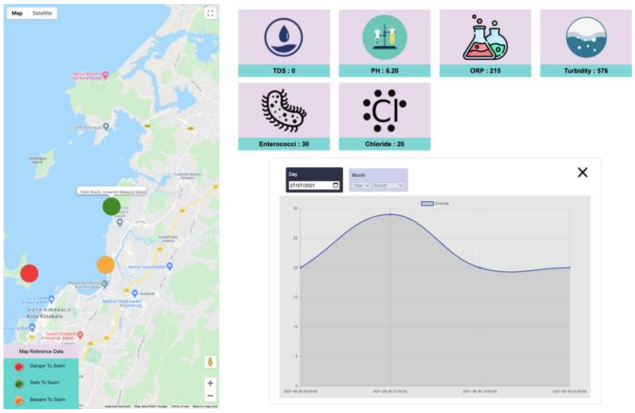
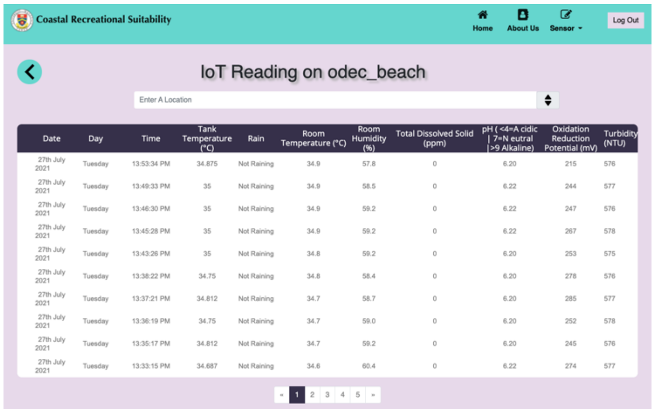


Fig. 10. The Page showing the Graph Representation of a Specific Variable.



IoT Reading on odec_beach

Date	Day	Time	Tank Temperature (°C)	Rain	Room Temperature (°C)	Room Humidity (%)	Total Dissolved Solid (ppm)	pH (<4=A cidic | 7=N eutral |>9 Alkaline)	Oxidation Reduction Potential (mV)	Turbidity (NTU)
27th July 2021	Tuesday	13:53:34 PM	34.875	Not Raining	34.9	57.8	0	6.20	215	576
27th July 2021	Tuesday	13:49:33 PM	35	Not Raining	34.9	58.5	0	6.22	244	577
27th July 2021	Tuesday	13:46:30 PM	35	Not Raining	34.9	59.2	0	6.22	247	576
27th July 2021	Tuesday	13:45:28 PM	35	Not Raining	34.9	59.2	0	6.22	267	578
27th July 2021	Tuesday	13:43:26 PM	35	Not Raining	34.8	59.2	0	6.20	253	575
27th July 2021	Tuesday	13:38:22 PM	34.75	Not Raining	34.8	58.4	0	6.20	278	576
27th July 2021	Tuesday	13:37:21 PM	34.812	Not Raining	34.7	58.7	0	6.20	285	577
27th July 2021	Tuesday	13:36:19 PM	34.75	Not Raining	34.7	59.0	0	6.20	252	578
27th July 2021	Tuesday	13:35:17 PM	34.812	Not Raining	34.7	59.2	0	6.20	245	576
27th July 2021	Tuesday	13:33:15 PM	34.687	Not Raining	34.6	60.4	0	6.22	274	577

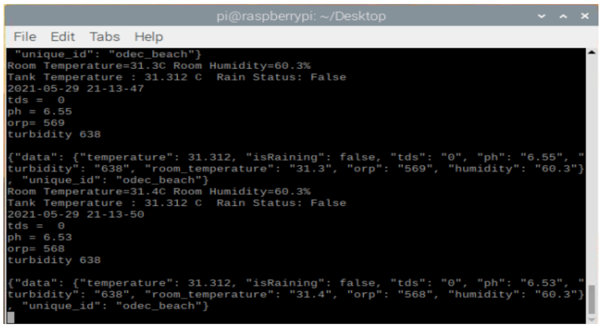
Fig. 11. The Page showing the IoT Readings of a Specific Beach.

V. Result and Discussion

A. Message Size

- Client Publish (Multi-Parameter)

Experiments are conducted to measure MQTT size for full system parameter message size. The script is run in Raspberry Pi that will publish the MQTT message to the broker. The python script will run for looping every second to transmit the message to the broker. The information is processed and can be viewed by subscribing to the message. Fig. 12 shows a script implemented on Raspberry Pi to publish a message to a broker.



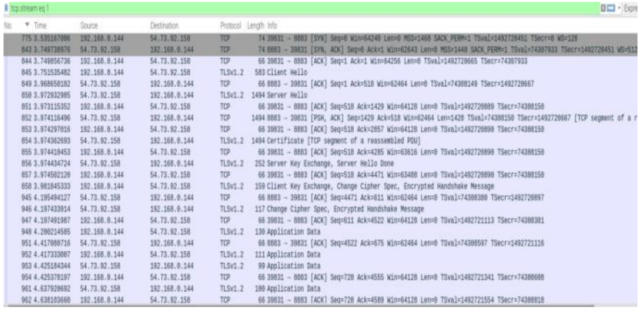
```
 "unique_id": "odec_beach"}
Room Temperature=31.3C Room Humidity=60.3%
Tank Temperature : 31.312 C  Rain Status: False
2021-05-29 21-13-47
tds =  0
ph = 6.55
orp= 569
turbidity 638

{"data": {"temperature": 31.312, "isRaining": false, "tds": "0", "ph": "6.55", "
turbidity": "638", "room_temperature": "31.3", "orp": "569", "humidity": "60.3"}
, "unique_id": "odec_beach"}
Room Temperature=31.4C Room Humidity=60.3%
Tank Temperature : 31.312 C  Rain Status: False
2021-05-29 21-13-50
tds =  0
ph = 6.53
orp= 568
turbidity 638

{"data": {"temperature": 31.312, "isRaining": false, "tds": "0", "ph": "6.53", "
turbidity": "638", "room_temperature": "31.4", "orp": "568", "humidity": "60.3"}
, "unique_id": "odec_beach"}
```

Fig. 12. Publish Message to the Broker.

Fig. 13 shows packets captured by Wireshark during transmission. It shows that the cloud broker used is HiveMQ are using Transport Layer Security V1.2 and it was captured using Wireshark. Before that, an MQTT broker is a server that receives all messages from the clients and then routes the messages to the appropriate destination clients. An MQTT client is any device (from a microcontroller up to a fully-fledged server) that runs an MQTT library and connects to an MQTT broker over a network. From a client to connect to a broker, it must run the TLSV1.2 encryption protocol to gain access with the MQTT broker. Fig. 13 also shows HiveMQ connections are in port 8883 which are more secured than 1883 because SSL will verify that the connection is encrypted using a certificate belonging to the domain name that you were trying to connect to. The two ends will also encrypt all traffic so that an observing party will be unable to eavesdrop. Without SSL IoT devices become vulnerable to a malicious broker impersonating the one you want to connect to. It is also possible that a third party might tamper with the data received. And an eavesdropper might attacker could see MQTT username and password.



```
No.  Time          Source         Destination    Protocol Length Info
775 3.535167006  192.168.0.144  54.73.92.158   TCP        74 39831 → 8883 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=1492720451 TSecr=0 WS=128
843 3.749738976  54.73.92.158   192.168.0.144  TCP        74 8883 → 39831 [SYN, ACK] Seq=0 Ack=1 Win=62643 Len=0 MSS=1440 SACK_PERM=1 TSval=74307933 TSecr=1492720451 WS=512
844 3.749856736  192.168.0.144  54.73.92.158   TCP        66 39831 → 8883 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=1492720665 TSecr=74307933
845 3.751535482  192.168.0.144  54.73.92.158   TLSv1.2   583 Client Hello
849 3.968658102  54.73.92.158   192.168.0.144  TCP        66 8883 → 39831 [ACK] Seq=1 Ack=518 Win=62464 Len=0 TSval=74308149 TSecr=1492720667
850 3.972932905  54.73.92.158   192.168.0.144  TLSv1.2  1494 Server Hello
851 3.973115352  192.168.0.144  54.73.92.158   TCP        66 39831 → 8883 [ACK] Seq=518 Ack=1429 Win=64128 Len=0 TSval=1492720889 TSecr=74308150
852 3.974116496  54.73.92.158   192.168.0.144  TCP      1494 8883 → 39831 [PSH, ACK] Seq=1429 Ack=518 Win=62464 Len=1428 TSval=74308150 TSecr=1492720667 [TCP segment of a r
853 3.974297016  192.168.0.144  54.73.92.158   TCP        66 39831 → 8883 [ACK] Seq=518 Ack=2857 Win=64128 Len=0 TSval=1492720890 TSecr=74308150
854 3.974362693  54.73.92.158   192.168.0.144  TLSv1.2  1494 Certificate [TCP segment of a reassembled PDU]
855 3.974418453  192.168.0.144  54.73.92.158   TCP        66 39831 → 8883 [ACK] Seq=518 Ack=4285 Win=63616 Len=0 TSval=1492720890 TSecr=74308150
856 3.974434724  54.73.92.158   192.168.0.144  TLSv1.2   252 Server Key Exchange, Server Hello Done
857 3.974502120  192.168.0.144  54.73.92.158   TCP        66 39831 → 8883 [ACK] Seq=518 Ack=4471 Win=63488 Len=0 TSval=1492720890 TSecr=74308150
858 3.981845333  192.168.0.144  54.73.92.158   TLSv1.2   159 Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
945 4.195494127  54.73.92.158   192.168.0.144  TCP        66 8883 → 39831 [ACK] Seq=4471 Ack=611 Win=62464 Len=0 TSval=74308380 TSecr=1492720897
946 4.197433914  54.73.92.158   192.168.0.144  TLSv1.2   117 Change Cipher Spec, Encrypted Handshake Message
947 4.197491987  192.168.0.144  54.73.92.158   TCP        66 39831 → 8883 [ACK] Seq=611 Ack=4522 Win=64128 Len=0 TSval=1492721113 TSecr=74308381
948 4.200214585  192.168.0.144  54.73.92.158   TLSv1.2   130 Application Data
951 4.417080716  54.73.92.158   192.168.0.144  TCP        66 8883 → 39831 [ACK] Seq=4522 Ack=675 Win=62464 Len=0 TSval=74308597 TSecr=1492721116
952 4.417333087  192.168.0.144  54.73.92.158   TLSv1.2   111 Application Data
953 4.425184344  54.73.92.158   192.168.0.144  TLSv1.2    99 Application Data
954 4.425378197  192.168.0.144  54.73.92.158   TCP        66 39831 → 8883 [ACK] Seq=720 Ack=4555 Win=64128 Len=0 TSval=1492721341 TSecr=74308608
961 4.637920692  54.73.92.158   192.168.0.144  TLSv1.2   100 Application Data
962 4.638103660  192.168.0.144  54.73.92.158   TCP        66 39831 → 8883 [ACK] Seq=720 Ack=4589 Win=64128 Len=0 TSval=1492721554 TSecr=74308818
```

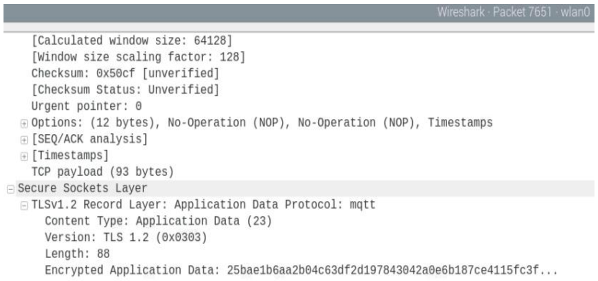
Fig. 13. Wireshark Packet Summary.

From observation has been made using Wireshark found total frame lengths are 297 bytes. Layer 1(Ethernet Frame) size is 14 bytes which contain source and destination IP addresses. Layer 2 (IP header) contains 20 bytes, Layer 3(TCP header) contains 32 bytes, Layer 4 (SSL Header) are 5 bytes. So, the total header involves 71 bytes. The encrypted application data contain 226 bytes per message transmitted. With a sum of 226 bytes and the header will contain exactly 297 bytes.

The data on encrypted applications contain parameters about Coastal Temperature, Humidity, Water Temperature, Rain Status, TDS, pH, ORP and turbidity and unique id(location). The data are sent using JSON format for example: {“data”: {“temperature”: “33.187”, “isRaining”: false, “tds”: “0”, “ph”: “6.10”, “turbidity”: “576”, “room_temperature”: “35.2”, “orp”: “310”, “humidity”: “51.8”}, “unique_id”: “odec_beach”}. To prove by combining all the parameters on a single packet could save bandwidth is by comparing the single message experiment on the next part.

- Client Publish (Single Parameter)

The next experiment uses a single parameter to send a message from the client to the broker. From the observation shown in Fig. 14, a single message packet contains a total of 159 bytes that contain ph value and identifier of location. For example, “data”: {“ph”: “6.48”}, {“unique_id”: “odec_beach”}. Layer 1 (Ethernet Frame) size is 14 bytes which contain source and destination IP addresses. Layer 2 (IP header) contains 20 bytes, Layer 3(TCP header) contains 32 bytes, Layer 4 (SSL Header contain 5 bytes for header and 88 bytes for the application data. In comparison, the header between single and multi-parameter are the same but has a huge difference on application data. Using multi-message that only contain 226 bytes from an average of 28.25 bytes per parameter. Thus, it shows three times better for saving network consumption and reducing complexity in MQTT system design.



```
    [Calculated window size: 64128]
    [Window size scaling factor: 128]
    Checksum: 0x50cf [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
  ⊞ Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
  ⊞ [SEQ/ACK analysis]
  ⊞ [Timestamps]
    TCP payload (93 bytes)
⊟ Secure Sockets Layer
  ⊟ TLSv1.2 Record Layer: Application Data Protocol: mqtt
      Content Type: Application Data (23)
      Version: TLS 1.2 (0x0303)
      Length: 88
      Encrypted Application Data: 25bae1b6aa2b04c63df2d197843042a0e6b187ce4115fc3f...
```

Fig. 14. Single Packet Parameter.

B. *Throughput*

Testing MQTT message throughput is important for an IoT application. It is also important to identify the maximum message throughput a single MQTT broker instance, as well as an elastically scaling MQTT broker cluster, can support without service degradation. This will help with capacity planning and can ensure deployment scale when the increase in usage.

The experiments are conducted to measure the throughput using ntopng installed on Raspberry Pi. It provides a web GUI to access accurate monitoring data. It provides detailed views on active hosts, flows, IP addresses, Mac addresses, Autonomous Systems, and throughput. From that, it can easily identify MQTT brokers and Pi. Throughput is how much information gets delivered in a certain amount of time. Most of the time network throughput is measured in bits per second (bps). The higher the throughput, the better the network and able to send data without packet drop. Fig. 15 explains that the address broker which is 54.73.92.158 are communicating with Raspberry Pi with IP address 192.168.0.144.

This experiment is conducted within three days. There are approximately 1400 messages sent every hour and it approximately generates 42,000 messages per day. The python code will run every second to collect sensor data. The result for three days analysis is shown in Table I. The result includes the date and time, duration, throughput (Raspberry Pi and Broker), network consumption (Raspberry Pi and Broker), and message size.

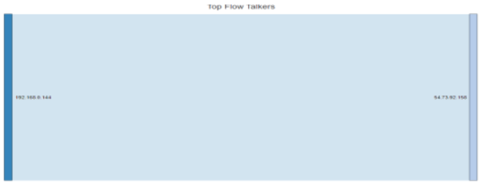


Fig. 15. Ntopng Flow Talker.

TABLE I. THROUGHPUT AND NETWORK USAGE

Date/ Time	Duration	RP T/P (kbit/s)	Broker T/P (kbit/s)	NC from RP (MB)	NC from Broker (MB)	Message Size (bytes)
24/7/2021 00:00:00	0	-	-	5.00	0.05793	296
12:30:00	30min	8.81	1.16	5.29	0.45426	297
01:00:00	1hr	10.49	3.17	9.64	0.98479	297
08:30:00	8.5hr	9.2	2.11	14.98	4.86	296
12:00:00	12hr	9.06	1.16	18.13	6.94	297
25/7/2021 00:00:00	24hr	7.61	1.74	30.77	14.21	296
12:00:00	36hr	5.63	2.11	43.75	21.55	297
26/7/2021 00:00:00	48hr	5.98	1.16	54.49	28.9	297
12:00:00	60hr	8.18	1.05	65.91	36.18	297
27/7/2021 00:00:00	72hr	9.13	3.17	77.15	43.62	297
Total	3 days	-	-	72.15 (minus 5.00MB)	43.62	-
Average	3 days	8.23	1.76	24.05MB/day	14.54MB/day	296.7

*Legend: T/P: Throughput, RP: Rasberry Pi, NC: Network Consumption.

Based on the table above, the throughput of Raspberry Pi is consistent between 5.63 kbit/s until 10.49 kbit/s with an average of 8.23 kbit/s. For MQTT broker throughput, it shows 1.05 kbit/s until 3.17 kbit/s with an average of 1.76 kbit/s. With the speed mentioned, it is considered average and consistent for sending 297 bytes packets to the broker. When testing on the ipantai system that connects with the broker, there is no delay of data. Therefore, the result supports MQTT as the suitable protocol for a system requiring less bandwidth, constrained devices, and high latency. The design principle is to minimise the network bandwidth and device resource requirements while attempting to ensure reliability and some degree of delivery assurance. However, in this research, there are no studies about other brokers that may outperform the investigated HiveMQ broker.

The network consumption depends on throughput. If there are more throughput in the network, more network consumption will be generated. For network consumption in Raspberry Pi, it consumes 72.15 MB during the three days observation. With that, an average of 24.05 MB bandwidth consumption per day is reasonable for an IoT based system to handle and maintain. If the device consumes 24.05 MB per day, it only consumes 721.5 MB per month. Thus, a user only needs to find an Internet plan with the said minimum requirement of 1GB quota per month. In this case, most providers in Malaysia offer around RM19 (USD4) per month/GB. It is affordable for developers to maintain. Since this project is using the beach as the focus of research for data gathering, the Internet bandwidth and coverage are not always available. However, MQTT serves as a lightweight protocol and keeps bandwidth requirements to an absolute minimum, handles unreliable networks and requires little implementation effort for developers. The message sizes are still consistent between 296 bytes and 297 bytes with 296.7 bytes as the average size within the three days. The message size depends on the reading. If there are more decimal places on the readings, more bytes need to be sent. For example, if the temperature increases from 31.0 to 35.456-degree Celcius, more bytes are being sent from the publisher.

The result shows better performance in terms of throughput and message sizes being transmitted compared to previous research [20][21][22]. The possible reason is different hardware [20] is using Wemos D1 R2 as microcontroller and Mosquitto as a broker, different message size [21] and using different cloud brokers such as Mosquitto and Paho [22]. MQTT performs better than AMQP in bandwidth usage and message throughput.

C. Power Consumption

This project uses the smart plug from Tenda for capturing and monitoring power consumption. The model used is Tenda Smart Wi-Fi Plug with Energy Monitoring (SP-9). Tenda SP-9 has included an energy monitoring function that can measure the power used by day/week/months/year. It can handle input from 100 V until 240 V between 50/60 Hz with 13A maximum load. To use the Smart Wi-Fi plug, a mobile device having Android 4.4 or higher or IOS 9.0 or higher is required. A WIFI network is also preferable. The experiment measured the power consumption of the broker and Raspberry Pi for three days. Table II shows an overview of the result taken and presented in the form of a table. The table shows the date and time, duration, current power (W) and power usage (kWh).

In Table II, it is found that the reading is consistent from 0.12kWh until 0.13kWh per day with an average of 0.126kWh. Next, the current power is used steadily from the range 5.26W until 5.65W with an average power of 5.44W. The power consumption of Raspberry Pi depends on the activity running on the operating system. If there is heavy activity such as streaming, the power consumption is significantly higher than usual. This result shows lower power consumption within the range of 0.12kWh to 0.13kWh per day. If it is multiplied by 30 days (1 month), it will consume approximately 6kW. The average electricity tariff per kWh in Malaysia is approximately RM0.32 (USD0.06). Therefore, it cost a minimum amount of about RM1.14 (USD0.30) per month. However, if the Raspberry Pi is powered up by solar, there should be no monthly cost for electricity which is possible for an IoT-based system. The research is that using Raspberry Pi and Arduino compared to another research using different hardware and sensor such as Arduino Uno and Waspmote [23] resulting in more power consumption while publishing data. Some other improvements in designing the IoT-based system may include security features using a goal-based approach [24], [25], explore suitable machine learning algorithms in water research [26], [27] and improve IoT networks using single-on with MQTT [28].

TABLE II. POWER CONSUMPTION (KWH)

Date/ Time	Duration	Current Power (W)	Power Usage (KWH)
24/7/2021 00:00:00 – 23:59:59	24 hours	5.26	0.12
25/7/2021 00:00:00 – 23:59:59	48 hours	5.41	0.13
26/7/2021 00:00:00 – 23:59:59	72 hours	5.65	0.13
Average	3 days	5.44	0.126

VI. CONCLUSION AND FUTURE WORK

In summary, the research has uncovered several significant discoveries. The objectives are to design and develop an IoT-based system for coastal recreational suitability systems and implement an IoT-based system to suit the requirements for capturing, storing, and reporting coastal environment data. The research presented an IoT based system design using MQTT protocol and conducted several experiments measuring the i) message size ii) throughput and iii) power consumption. The result has shown the specified design and system development benefits from using MQTT as the primary protocol of communication between IoT devices. The message size is captured in small packets, which means less data consumption and power required than certain protocols, such as HTTP. The smaller message size makes it ideal for situations where bandwidth is limited, which is an important aspect of IoT devices. The next finding obtained from this research is better performance was presented using HiveMQ broker as part of the configuration compared to previous research. The throughput captured by HiveMQ cloud brokers is faster to send and receive data efficiently. This has also shown that MQTT is lightweight, quick, energy-efficient for a system, which is crucial for the implementation of IoT. Therefore, these findings summarise that the HiveMQ cloud broker has several advantages and is suitable for coastal recreational configuration. As this setup may involve many locations, a cloud broker is suitable and possible future work may need to centralise the connections between many places.

Finally, the findings hope to promote MQTT as the main communication specifically in the industry 4.0 (IR4.0) era. IR 4.0 could enable technologies and a new dimension in environmental monitoring systems to preserve and provide a safer environment for people to undertake coastal recreational activities. In the future, these systems can provide real-time data, decentralise analytics and support decision making.

ACKNOWLEDGMENT

This work is supported in part by the Universiti Malaysia Sabah under Grant DN20081. The authors would like to thank the research assistants for their assistance in the state-of-art analysis work the anonymous reviewers for their comments.

REFERENCES

[1] B. Breen, J. Curtis, and S. Hynes, “Water quality and recreational use of public waterways,” J. Environ. Econ. Policy, vol. 7, no. 1, pp. 1–15, 2018, doi: 10.1080/21606544.2017.1335241.

[2] S. Dorevitch, S. DeFlorio-Barker, R. M. Jones, and L. Liu, “Water quality as a predictor of gastrointestinal illness following incidental contact water recreation,” Water Res., vol. 83, pp. 94–103, 2015, doi: 10.1016/j.watres.2015.06.028.

[3] L.-H. Lee and Y.-D. Lee, “The impact of water quality on the visual and olfactory satisfaction of tourists,” Ocean Coast. Manag., vol. 105, pp. 92–99, 2015, doi: https://doi.org/10.1016/j.ocecoaman.2014.12.020.

[4] F. I. Salih, N. A. Abu Bakar, N. H. Hassan, F. Yahya, N. Kama, and J. Shah, “IoT Security Risk Management Model for Healthcare Industry,” Malaysian J. Comput. Sci., vol. 0, no. 0 SE-Articles, pp. 131–144, Dec. 2019, doi: 10.22452/mjcs.sp2019no3.9.

[5] M. M. Rashid, A.-A. Nayan, S. A. Simi, J. Saha, Md. Obaidur Rahman, and M. G. Kibria, “IoT based Smart Water Quality Prediction for Biofloc Aquaculture,” Int. J. Adv. Comput. Sci. Appl., vol. 12, no. 6, pp. 56–62, 2021.

[6] E. C. Heagney, J. M. Rose, A. Ardeshiri, and M. Kovač, “Optimising recreation services from protected areas – Understanding the role of natural values, built infrastructure and contextual factors,” Ecosyst. Serv., vol. 31, pp. 358–370, 2018, doi: 10.1016/j.ecoser.2017.10.007.

[7] J. H. Nord, A. Koohang, and J. Paliszkiewicz, “The Internet of Things: Review and theoretical framework,” Expert Syst. Appl., vol. 133, pp. 97–108, 2019, doi: 10.1016/j.eswa.2019.05.014.

[8] S. Luthra, Garg.D, S. K. Mangla, and Y. P. Singh Berwal, “Analyzing challenges to Internet of Things (IoT) adoption and diffusion: An Indian context,” Procedia Comput. Sci., vol. 125, pp. 733–7399, 2018.

[9] M. Asplund and S. Nadjm-Tehrani, “Attitudes and Perceptions of IoT Security in Critical Societal Services,” IEEE Access, vol. 4, pp. 2130–2138, 2016, doi: 10.1109/ACCESS.2016.2560919.

[10] M. H. Miraz, M. Ali, P. S. Excell, and R. Picking, “A Review on Internet of Things (IoT), Internet of Everything (IoE) and Internet of Nano Things (IoNT),” Sep. 2017, doi: 10.1109/ITechA.2015.7317398.

[11] E. A. Kosmatos, N. D. Tselikas, and A. C. Boucouvalas, “Integrating RFIDs and Smart Objects into a UnifiedInternet of Things Architecture,” Adv. Internet Things, vol. 01, no. 01, pp. 5–12, 2011, doi: 10.4236/ait.2011.11002.

[12] T. Al-Rousan, “The Future of the Internet of Things,” Int. J. Comput. Commun. Control, vol. 4, Apr. 2017, doi: 10.15242/IJCCIE.AE0417133.

[13] R. Girau, E. Ferrara, M. Pintor, M. Sole, and D. Giusto, Be Right Beach: A Social IoT System for Sustainable Tourism Based on Beach Overcrowding Avoidance. 2019.

[14] D. Rahmawati, N. I. Arvitrida, D. Lastomo, Kusnadi, and Rindawati, “Smart economy for coastal resource management in Surabaya City,” {IOP} Conf. Ser. Earth Environ. Sci., vol. 340, p. 12016, Oct. 2019, doi: 10.1088/1755-1315/340/1/012016.

[15] G. Xu, Y. Shi, X. Sun, and W. Shen, “Internet of Things in Marine Environment Monitoring: A Review,” Sensors, vol. 19, no. 7, 2019, doi: 10.3390/s19071711.

[16] R. Khan, S. U. Khan, R. Zaheer, and S. Khan, “Future Internet: The Internet of Things Architecture, Possible Applications and Key Challenges,” in 2012 10th International Conference on Frontiers of Information Technology, 2012, pp. 257–260, doi: 10.1109/FIT.2012.53.

[17] WHO, “WHO | Guidelines for safe recreational water environments,” vol. 1, pp. 118–127, 2013.

[18] K. Mekki, E. Bajic, F. Chaxel, and F. Meyer, “A comparative study of LPWAN technologies for large-scale IoT deployment,” ICT Express, vol. 5, no. 1, pp. 1–7, 2019, doi: https://doi.org/10.1016/j.icte.2017.12.005.

[19] H. A. Kusuma, R. Anjasmara, T. Suhendra, H. Yunianto, and S. Nugraha, “An IoT Based Coastal Weather and Air Quality Monitoring Using GSM Technology,” J. Phys. Conf. Ser., vol. 1501, no. 1, 2020, doi: 10.1088/1742-6596/1501/1/012004.

[20] D. Eridani, K. T. Martono, and A. A. Hanifah, “MQTT performance as a message protocol in an IoT based chili crops greenhouse prototyping,” 2019 4th Int. Conf. Inf. Technol. Inf. Syst. Electr. Eng. ICITISEE 2019, vol. 6, pp. 184–189, 2019, doi: 10.1109/ICITISEE48480.2019.9003975.

[21] M. A. Triawan, H. Hindersah, D. Yolanda, and F. Hadiatna, “Internet of Things using publish and subscribe method cloud-based application to NFT-based hydroponic system,” Proc. 2016 6th Int. Conf. Syst. Eng. Technol. ICSET 2016, pp. 98–104, 2017, doi: 10.1109/FIT.2016.7857546.

[22] M. Pohl, J. Kubela, S. Bosse, and K. Turowski, “Performance evaluation of application layer protocols for the internet-of-things,” Proc. - 2018 6th Int. Conf. Enterp. Syst. ES 2018, pp. 180–187, 2018, doi: 10.1109/ES.2018.00035.

[23] M. Pavelic, V. Bajt, and M. Kusek, “Energy efficiency of Machine-to-Machine protocols,” pp. 361–366, 2018.

[24] F. Yahya, R. J. Walters, and G. B. Wills, “Goal-based security components for cloud storage security framework: A preliminary study,” 2016 Int. Conf. Cyber Secur. Prot. Digit. Serv. Cyber Secur. 2016, 2016, doi: 10.1109/CyberSecPODS.2016.7502338.

[25] F. Yahya, V. Chang, R. J. Walters, and G. B. Wills, “A security framework to protect data in cloud storage,” IoTBDS 2019 - Proc. 4th Int. Conf. Internet Things, Big Data Secur., no. IoTBDS, pp. 307–314, 2019, doi: 10.5220/0007737603070314.

[26] F. Yahya, B. Fazli, H. Sallehudin, and M. I. M. Jaya, “Machine learning in dam water research: An overview of applications and approaches,” Int. J. Adv. Trends Comput. Sci. Eng., vol. 9, no. 2, pp. 1268–1274, 2020, doi: 10.30534/ijatcse/2020/56922020.

[27] B. M. Fazli et al., “Improvement of Dam Management in Terms of WAM Using Machine Learning,” in In: Mohd Sidek L., Salih G., Boosroh M. (eds) ICDSME 2019. ICDSME 2019. Water Resources Development and Management, 2020, pp. 226–236, doi: https://doi.org/10.1007/978-981-15-1971-0_23.

[28] L. Nguyen, T. Thanh, N. N. Phien, T. A. Nguyen, and H. K. Vo, “UIP2SOP : A Unique IoT Network applying Single Sign-On and Message Queue Protocol,” Int. J. Adv. Comput. Sci. Appl., vol. 12, no. 6, 2021.