Comparison of IoT Messaging Protocols: A novel Crop-Specific Protocol for Wheat, Banana, and Chili

Komal Memon[1], Fahim Aziz Umrani[1], Attiya Baqai[1], Zafi Shehran Shah[1]
[1]Mehran University of Engineering and Technology Jamshoro
*Correspondence: komal.memon@admin.muet.edu.pk

Citation | Memon. K, Umrani. F. A, Baqai. A, Shah. Z. S, "Comparison of IoT Messaging protocols: A novel crop specific protocol for Wheat, Banana, and Chili", IJIST, Vol. 06 Issue. 04 pp 2225-2238, Dec 2024
Received | Nov 28, 2024 **Revised |** Dec 20, 2024 **Accepted |** Dec 26, 2024 **Published |** Dec 28, 2024.

IoT systems mostly depend on messaging protocols to facilitate the exchange of IoT data, with various protocols or frameworks available to support different types of messaging patterns. Choosing a suitable IoT messaging protocol for a specific application is a significant task. It is crucial to select a protocol that meets criteria such as reliability, lightweight, scalability, extensibility, interoperability, and security. It is crucial to opt for a protocol that meets criteria such as being reliable, lightweight, scalable, extensible, interoperable, and secure. With the increasing prevalence of machine-to-machine communication, numerous standardized communication protocols have emerged for IoT applications. However, performance characteristics of IoT protocols can vary expressively, even when operating under the same conditions. This research paper presents a quantitative comparison of three well-known IoT messaging protocols: MQTT (Message Queuing Telemetry Transport), AMQP (Advance Message Queuing Protocol), and HTTP (Hypertext Transfer Protocol). This research focuses on comparison of existing protocols to latency and throughput. A novel crop-specific protocol is also designed for wheat, Banana, and chili crops.

Keywords: Messaging Protocols, MQTT, AMQP, HTTP, Crop Specific.

Introduction:

Precision Agriculture reshaped traditional agriculture by introducing the Internet of Things (IoT). It advances precision agriculture and reshaped the creation of IoT devices, its way of communicating and interpret data. The transformation becomes increasingly obvious as IoT devices churn out data at an increasing rate. The inclusion of IoT into traditional agriculture provides a transformative approach which is called precision agriculture (PA). This technology uses sensors with other connected devices to transform traditional farming into smart farming [1]. In IoT-based systems, messaging protocols play a critical role when connected devices exchange data. These protocols serve as the core foundation for establishing connectivity between diverse and heterogeneous devices. As a result, a wide range of messaging protocols has emerged, enabling more efficient data exchange among IoT devices [2][3][4][5].

There are five layers for IoT-enabled devices: the first is the business layer, followed by the application layer, the middleware layer, the network layer, and finally the perception layer (refer to Figure. 1). These layers play a crucial role in the connectivity and functionality of IoT devices.

- The business layer is responsible for system management.
- The application layer manages system applications and facilitates data transmission and reception.
- The middleware layer processes information, performs actions, and stores data in databases.
- The network layer ensures secure data transmission.
- The perception layer is responsible for storage, information processing, and actions [6] [7].

Since the application layer primarily handles data transmission and reception, this research focuses on messaging protocols related to the application layer. Different application layer protocols are available for data transmission and reception such as HTTP, MQTT, AMQP, Data Distribution service (DDS), Constrained Applications (CoAP), Extensible Messaging and Presence Protocol (XMPP) and many others [8]. These protocols are different in terms of connectivity options but also some similarities are present. Therefore, it is very challenging to choose which protocol is suitable for the specific application. The selection of proper messaging protocol is very important for the overall performance of the system or IoT devices, it directly impacts factors such as data transmission efficiency, reliability, latency, and power consumption, ultimately influencing the effectiveness and scalability of the smart farming network. Proper selection of protocols reduces the traffic intensity and latency and enhances the reliability of the system [9]. The choice of an appropriate messaging protocol depends on several factors, including the business requisites of the IoT application, software capabilities, device or hardware limitations, typical data exchange size, network reliability, latency requirements, security considerations, scalability needs, and power consumption constraints, ensuring that the protocol aligns with the specific goals and constraints of the system. When deploying IoT devices and developing IoT applications, it is paramount to consider the distinctive characteristics and unique functionalities offered by various existing protocols [10][11]. This study focuses on the comparison of HTTP, AMQP, and MQTT protocols. Also, it highlights crop-specific protocols for three main crops Wheat(grain), Chillies (Vegetable), and Banana(fruit).

The structure of this research paper is organized as follows:

- **Section II**: Related Work – Explores previous studies and research on IoT messaging protocols.
- **Section III**: Research Novelty – Presents the novel contributions of this research.
- **Section IV**: Comparative Analysis of IoT Application Layer Protocols.
- **Section V**: Relative Comparison of Application Layer Protocols.
- **Section VI**: Need for Crop-Specific Protocols for Three Crops.

- **Section VII**: Designing Crop-Specific Protocols.
- **Section VIII**: Conclusion – Summarizes the findings and provides recommendations for future research.

Literature Review:

Research and exploration within the IoT field have predominantly concentrated on data collection, analysis, transformation, and collaborative processes carried out through RESTful services. To design the flow of IoT data, an IT reference model is utilized, encompassing multiple layers: (1) Devices and Controllers (Perception Layer), (2) Connectivity (Network Layer), (3) Data Aggregation and Abstraction (Middleware Layer), (4) Application Layer, and (5) Business Layer**.** In this context, we recognize significant studies that focus on the application layer. In [2], authors carried out research based on experiments for six various messaging protocols, analyzing their advantages and disadvantages. The survey aims to provide an understanding of the pros and cons of these protocols. In [3][12], a comparative study is being performed by the authors to evaluate the applicability of messaging protocols such as COAP and MQTT for IOT-based healthcare applications. The research paper specifically investigates the fragility of these protocols and their security imputation within the healthcare domain. The study places a particular emphasis on security aspects, aiming to identify vulnerabilities and flaws in safeguarding sensitive and personal data related to patients. The three main threats identified by the authors are categorized as:(a) Privacy & Confidentiality (b)Availability (c)Integrity. In [4][13], the researchers provide a comprehensive overview of IoT architecture, examining its fundamental aspects. Additionally, they extensively discuss the communication protocols specifically designed for IoT technology. Furthermore, the study includes an analysis of security threats and common implementation challenges, along with an exploration of various sectors that can greatly benefit from the advancements in IoT development. This paper focuses on evaluating the performance of IoT communication protocols at the application layer, specifically AMQP, CoAP, and MQTT. The study assesses these protocols based on metrics such as throughput, message size, and packet loss [5]. This study introduces models of MQTT, MQTT-SN, and CoAP protocols and focuses on verifying various communication properties associated with these protocols. The communication properties examined include connection establishment, persistent sessions, caching, proxying, message ordering, and quality of service (QoS). These protocol models are designed to be integrated with or extended to other formal models of IoT systems using machine decomposition within the Event-B framework [6][9][10]. In [14] researchers investigate IOT protocols and highlight the advantages and disadvantages. In [15][16], researchers compare the HTTP and MQTT protocols in a way that latency should be low and a device becomes modular. The hardware stage used in the experiments outlined in the article is based on the Esp32 platform, with programming language C employed for implementation. The paper emphasizes the significance of selecting the appropriate IoT protocol that aligns with the purpose, hardware, and software components of the system. Given the diverse requirements across various IoT use cases, a wide range of IoT protocols are available to address specific needs and cater to different scenarios. The research [17][18][19] indicates the comparison of different application layer protocols and analyzed performance in various scenarios such as security, lightweight, and fast data transfer.

Research Novelty and Objectives:

This research aims to compare IoT messaging protocols such as HTTP, AMQP, and MQTT. The comprehensive comparison of three different protocols during the IoT device connections and measure the performance metrics such as latency and throughput and suggests the suitable protocol for IoT devices to communicate, where efficient and consistent data exchange is critical for optimizing agricultural processes and enhancing productivity.

A novel crop-specific protocol has been designed for wheat, chilies, and banana crops in Python to help farmers and gardeners. It contributes to precision agriculture and crop

management systems. The dual focus on comparison as well as designing of crop-specific protocols shows the research contribution.

Messaging Protocols for IoT:

The three main messaging protocols that are widely used for IoT devices are MQTT, AMQP, and HTTP. This section is comprised of detailed discussions regarding IoT application layer protocols.

MQTT (Message Queuing Telemetry Transport Protocol):

Message Queuing Telemetry Transport (MQTT) is a publish-subscribe (Pub-Sub) messaging protocol. It is a lightweight, Organization for the Advancement of Structured Information Standards (OASIS) standard protocol [13]. It is suitable for real-time data transfer applications such as precision agriculture. MQTT was developed to facilitate lightweight Machine-to-Machine (M2M) communication in networks with constrained resources [20][21]. In the MQTT architecture, clients act as publishers and subscribers of messages, while an MQTT broker serves as an intermediary for message exchange as shown in Figure 2. Published messages can be retained by the broker for future subscriptions. Each message in MQTT is linked to a specific topic, allowing clients to subscribe to these topics and thereby receive all messages published on them. MQTT, being a binary protocol, typically employs a fixed 2-byte header and can accommodate message payloads of varying sizes, ranging from small data packets to a maximum of 256 MB. MQTT provides secure communication using TCP protocol. This protocol offers three levels of Quality of service (QoS) QoS 0, QoS 1, and QoS 2 for ensuring message delivery. MQTT also known as fundamental messaging protocol mostly used in IOT devices [22][23].

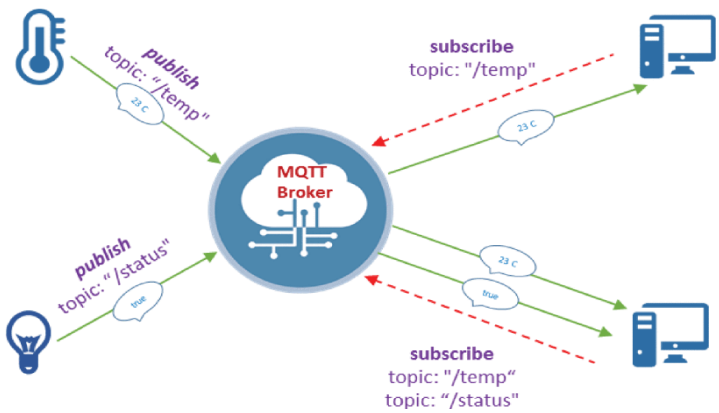

Figure 2: MQTT Pub-sub model [2]

In the Pub-sub mechanism publisher sends the single command to the broker, while the broker sends or broadcasts that message to all subscribers associated with the "topic". As in Figure 2 publisher is publishing the topic "temperature" and "status" to the MQTT broker and that broker is broadcasting the temperature to the subscribers [24]. Whereas the client must subscribe to the "topic" for receiving data.

Advanced Message Queuing Protocol (AMQP):

Advanced Message Queuing Protocol (AMQP) is a lightweight M2M protocol. It is both ISO and OASIS standard protocol [25][26]. To tackle interoperability issues AMQP protocol is used. It supports a queuing mechanism. Just like MQTT, AMQP also supports the pub-sub mechanism and consumers have to subscribe to the topic. AMQP enables both publish-subscribe and request-response models based on topics. It does not broadcast messages but sends messages to specific consumers who have subscribed to specific topics. AMQP goes further than MQTT by supporting business transactions and providing additional features such as flexible routing [27]. In an AMQP mechanism, first consumer or publisher creates

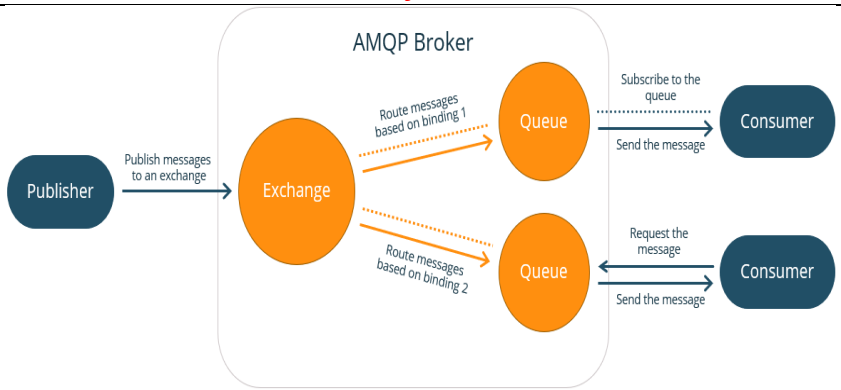

Figure 3: AMQP model [25]

An 'exchange' with a specific name is created and then broadcast. This exchange name is used to connect consumers and publishers to each other. The consumer creates a 'queue' and binds it to the exchange. To bind the exchange, received messages should be routed to a matching or appropriate queue.

- **Broker**: It manages exchanges, delivery, and queues of messages.
- **Exchange**: It has the binding key and routing key. As messages are received by a publisher it directs threat-received messages to the queues by using binding and routing keys.
- **Queue:** It receives messages from exchanges and stores them according to queuing rules.
- **Consumer:** It subscribes to the queues and receives or processes the messages.

In large-scale agriculture, AMQP provides centralized data management where Sensors (e.g., soil moisture, temperature, humidity, and pH sensors) and devices publish data to a centralized AMQP message broker. The broker routes this data to appropriate systems, such as cloud platforms, databases, or local controllers. It provides reliable communication, flexible data routing, Interoperability, security, and real-time data and control.

Hypertext Transfer Protocol:

HTTP, initially conceived by Tim Berners-Lee, emerged as a preeminent web messaging protocol. Following collaborative refinement by the IETF and W3C, it achieved standardization in 1997 [18]. Rooted in a request/response structure, HTTP underpins RESTful Web architecture. Analogous to CoAP, it employs Universal Resource Identifiers (URIs) in place of topics. Data transmission transpires via servers addressing URIs, while clients retrieve data through specific URIs. Operating as a text-based protocol, HTTP avoids dictating header and message payload sizes, deferring instead to web server or programming technology choices. Notably, TCP serves as its default transport mechanism, complemented by TLS/SSL for bolstered security [15][28]. In this manner, interactions between clients and servers adopt a connection-oriented approach. While lacking an explicit Quality of Service (QoS) definition, HTTP can incorporate supplementary measures for QoS enforcement. As a globally embraced web messaging standard, HTTP offers an array of attributes including persistent connections, request pipelining, and chunked transfer encoding. These facets facilitate efficient and responsive communication.

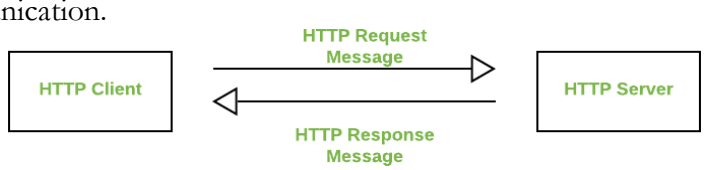

Figure 4: HTTP Model

Comparative Analysis of Iot Messaging Protocols:

This section introduces the comprehensive comparative examination of three prominent IoT messaging protocols within IoT systems: MQTT, AMQP, and HTTP. The analysis is structured around multiple criteria, offering an informative distinctive attribute. The comprehensive findings of this comparative investigation are shown in Table I, providing a consolidated reference for easy cross-evaluation.

Table.1. Analysis of Messaging Protocols

Factors	HTTP	MQTT	AMQP
Quality of Service	TCP	QoS 0, QoS 1, QoS 2	Settle/unsettle
Port	80	1883	5671
Message Size	Heavyweight	Lightweight	Lightweight
Architecture	Client/Server	Client/Broker	Client/Broker Client/Server
Keywords	Request/Response	Publisher/Subscriber	Request/Response Publisher/Subscriber
Python Library	HTTP	Paho	Pika
Connection	1-1	1-1, 1-N, N-N	Point to point
Data Received	High	Medium	Low
Throughput	Medium	High	Low
Latency	High	Low	Medium
Standards	IETF	Eclipse	IETF
License	Free	Open source	Open source
Application	Web	IoT Automation	Business messaging

HTTP:

HTTP is Suitable for applications where devices occasionally send data (e.g., a weather station uploading hourly data). In agriculture HTTP is used where applications that involve periodic data uploads or integration with web-based systems (e.g., dashboards or reports).

MQTT:

It is Lightweight, with minimal bandwidth and power consumption, Ideal for real-time applications like triggering irrigation systems based on soil moisture levels. In agricultural IoT, MQTT is used where applications require real-time data exchange between sensors and actuators (e.g., greenhouse climate control).

AMQP:

Advanced features like message queuing, priority, and routing via exchanges. It requires Higher computational and memory requirements compared to MQTT. In agricultural IoT, it is applicable in Large-scale systems where data from many sensors must be routed to different endpoints (e.g., weather predictions to the cloud, irrigation commands to actuators, and alerts to farmers' mobile devices).

Relative Analysis of Protocols:

In this segment, a thorough and contextual evaluation of the three IoT messaging protocols, namely AMQP, MQTT, and HTTP. The examination is focused on two closely interconnected criteria, thereby elucidating the strengths and limitations inherent in each messaging protocol. These protocols are notably comprehensive and divergent due to their distinct evolution processes and requisites. Furthermore, their precise and relative comparisons are contingent upon various factors such as IoT system classifications, device specifics, resource allocations, application scenarios, and contextual exigencies [29]. The dynamics of IoT components may yield different comparative outcomes, distinct from those presented here. Moreover, this assessment is on stationary modules and draws from empirical evidence found in the literature. Notably, it does not encompass the dynamic intricacies of network conditions

or the overheads associated with packet retransmissions. These factors can significantly influence comparative results and introduce variations.

Throughput:

The throughput of IoT messaging protocols refers to the rate or frequency at which data can be sent or processed between transmission and reception devices, clients, or servers. It is a critical performance metric that indicates how efficiently and quickly messages can be exchanged within an IoT system. Different messaging protocols exhibit varying levels of throughput due to their design, features, and underlying technologies. Table 2 compares the performance of HTTP, AMQP, and MQTT protocols based on latency and throughput across varying message loads (10 to 50 requests). MQTT demonstrates the best performance, achieving the lowest latency (0.12–0.59 seconds) and highest throughput (87.87–97.3 messages/sec), making it ideal for real-time, high-speed applications like IoT. AMQP shows moderate performance, with latency ranging from 0.9 to 1.33 seconds and throughput improving from 10.74 to 37.4 messages/sec, highlighting its scalability for higher message loads. HTTP, while consistent in throughput (54–58.41 messages/sec), suffers from increasing latency (0.22–1.05 seconds), making it less suitable for time-sensitive tasks. Overall, MQTT outperforms the others in both efficiency and responsiveness. In Figure 5 it can be analyzed that the throughput of MQTT is higher than AMQP and HTTP.

Table 2. Evaluation of HTTP, AMQP, and MQTT: Latency vs Throughput

Request (Messages)	HTTP Latency	HTTP Throughput	AMQP Latency	AMQP Throughput	MQTT Latency	MQTT Throughput
10	0.22	54.7	0.9	10.74	0.12	90.09
20	0.43	57.47	1.01	19.7	0.22	97.3
30	0.66	55.75	1.12	26.7	0.36	88.2
40	0.85	57.56	1.24	32.25	0.44	95.2
50	1.05	58.41	1.33	37.4	0.59	87.87

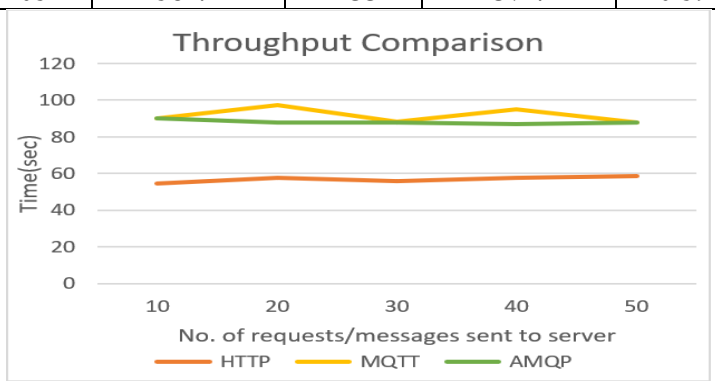

Figure 5: Throughput Comparison of IOT Messaging Protocols

Latency:

Latency can be defined as the delay that occurs while sending a message until receiving a response. In IoT systems, low latency is often crucial, especially for real-time applications and control systems. Each of these messaging protocols has different characteristics when it comes to latency. Whereas in Figure 6 the analysis shows that MQTT is performing much better in terms of latency from HTTP and AMQP.

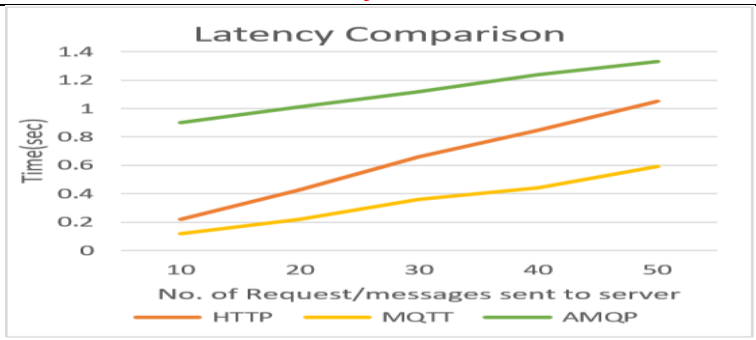

Figure 6: Latency Comparison of IOT Messaging Protocols

Need of Crop-Specific Protocols and Limitations of IOT Messaging Protocols:

In agriculture, IoT messaging protocols face certain shortcomings that can impede their effectiveness in addressing the unique challenges of the industry. One notable issue is the high-power consumption associated with some generic IoT protocols, which is especially problematic in agricultural settings where many devices operate on limited battery capacity. Moreover, the limited connectivity prevalent in rural areas can hinder the seamless functioning of certain protocols, affecting data transmission and overall efficiency. Additionally, generic IoT protocols may transmit excessive data, leading to higher data costs and potentially overwhelming the limited network bandwidth available in remote farming regions. Furthermore, security and privacy concerns arise as some protocols may lack robust measures to protect sensitive agricultural data, leaving valuable information vulnerable to cyber threats. Finally, the lack of standardization among IoT protocols poses interoperability challenges, making it difficult to integrate various devices from different manufacturers effectively. To address these issues, there is a growing need for crop-specific IoT protocols tailored to the unique requirements of agriculture. Crop-specific protocols can be designed to transmit only the essential data relevant to specific crop types, eliminating unnecessary information and reducing data overhead. Moreover, custom protocols can be optimized to work efficiently with low-power IoT devices commonly used in agriculture, ensuring prolonged device operation without frequent battery replacements. By catering to the connectivity limitations of rural areas, crop-specific protocols can facilitate more reliable data transmission and communication between devices. Additionally, these protocols can be equipped with enhanced security measures, safeguarding sensitive agricultural data and protecting against potential cyber threats. Lastly, crop-specific protocols can be standardized to promote interoperability, allowing seamless integration of diverse agricultural devices and systems. As the agricultural industry increasingly embraces IoT technologies, the development and implementation of crop-specific protocols hold great potential in maximizing efficiency, sustainability, and productivity in farming practices.

Transmission of Crop-Specific Data Using IoT Messaging Protocols:

In agricultural IoT, transmitted data is related to soil parameters such as soil moisture, temperature, humidity, and pH. The crop-specific data is firstly collected from sensors and then it is transmitted to the cloud using IoT messaging protocols. Each protocol has its distinctive rules for transmitting data. Some of them are listed below:

HTTP:

In HTTP sensors send crop-specific data using HTTP POST to a central server or cloud platform. From where clients easily visualize or analyze data through web dashboards or mobile applications.

MQTT:

In MQTT sensors publish crop-specific data such as soil moisture, pH, and other parameters to a broker using specific topics, such as farm/soil/moisture. Whereas clients such as farmer mobile apps subscribe to relevant topics to receive updates in real-time. This is a phenomenon of MQTT for crop-specific data transmission.

AMQP:

In AMQP crop-specific data is sent to an AMQP broker (e.g., RabbitMQ) and routed to appropriate queues based on predefined rules or bindings. Whereas Farmer's applications or dashboards consume data from the queues as needed.

Table 3. Comparison for crop-specific data

Factors	HTTP	MQTT	AMQP
Data flow	On-demand	Real-time data	Buffered
Data Transmitted	Drone-based crop health	Soil moisture, pH, humidity	Complex or high-volume data
Best Feature	Data uploads periodically	Lightweight real-time data	Large-scale data upload
Complexity	low	low	High

The factors mentioned in Table 3 favor the MQTT protocol for crop-specific data transmission in an agricultural system, whereas the combination of all three protocols provides layered data transmission.

Novel Crop-Specific Protocols for Wheat, Banana, and Chili:

The novel crop-specific protocol has been designed in Python coding based on the soil pH, Temperature, humidity, and Moisture parameters investigated in research, further, these parameters have been used to design an algorithm that recommends season, fertilizers, length of stay, and water period for a specific crop.

Table 4. Optimum values of three crops

Crop name	Soil pH	Air Temperature(C)	Soil Humidity	Soil Moisture	Season	Fertilizer	Water Period	Yield Per Acre
Banana [30][31]	5.5-7	26-30	45.33%	70%	Tropical	NPK (potassium rich)	7-10 days	10-12 tons
Wheat [32]	6-7.5	15-25	50-70%	60-80%	Rabi (winter)	N, P, K	10-14 days	1.5-2.5 tons
Chili [33][34]	5.5-7	20-30	70%	19.8%	Kharif/ Rabi	N, P, K	5-7 days	1.5-2 tons

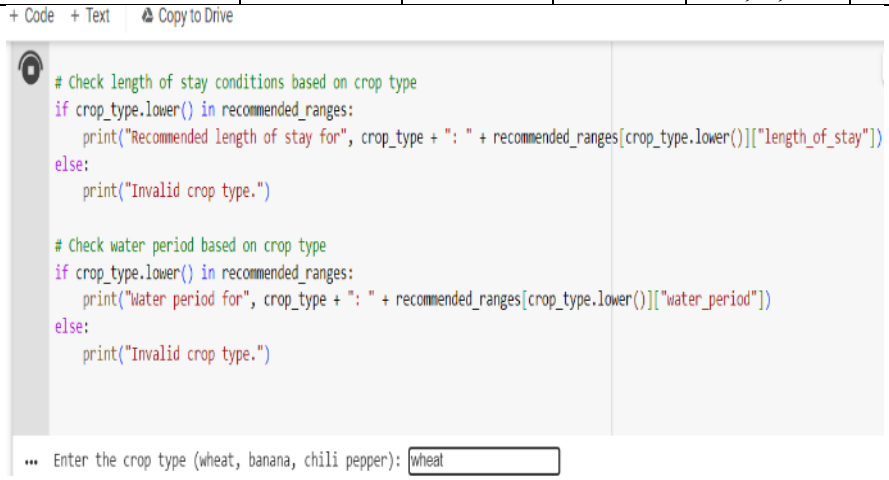

Figure 7: Prompt to enter crop type

Crop-specific Protocol Algorithm and Flow Chart:

Step 1: Start

Step 2: Input crop type

Step 3: Validate the crop
- If crop type is invalid → Go to Step 4
- If crop type is valid → Go to Step 5

Step 4: End
Step 5: For a valid crop, check specific crop details:
- If crop type = "Wheat Crop":
- Retrieve Fertilizer recommendations
- Retrieve Crop length
- Retrieve Soil parameters

Step 6: Provide water period recommendations
Step 7: Retrieve fertilizer details
Step 8: Display the recommendation results
Step 9: End

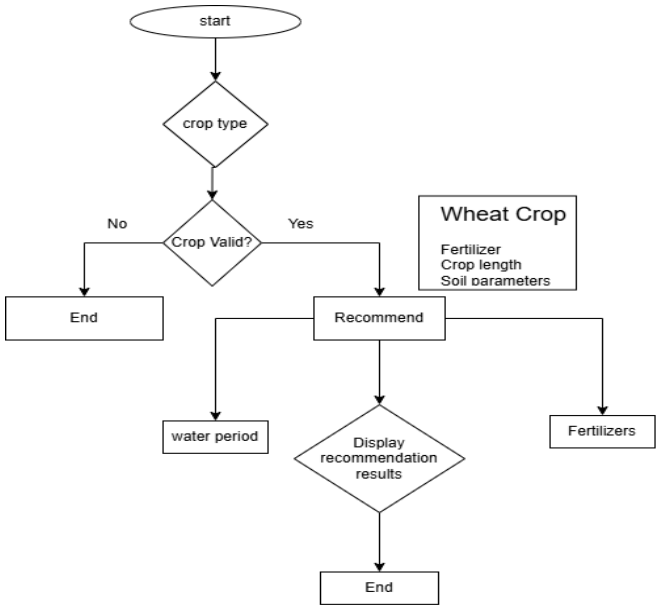

Figure 8: Flow chart for crop-specific protocol

This flowchart outlines a systematic approach to guide farmers in managing crop-specific requirements. It begins with the identification of the *crop type that which crop has been entered by the user*, which is then validated to ensure its compatibility with the given conditions. If the crop is deemed invalid, the process ends. For valid crops, such as wheat, the system progresses to the *recommendation* stage, where critical parameters like fertilizers, water periods, and soil conditions are analyzed. These recommendations are displayed to the user, providing detailed insights such as suitable fertilizers, watering schedules, and growth management techniques. This process ends with the *display of recommendation results*, ensuring that farmers receive tailored guidance to optimize crop productivity. This structured flow enhances decision-making, reduces resource wastage, and supports precision agriculture practices.

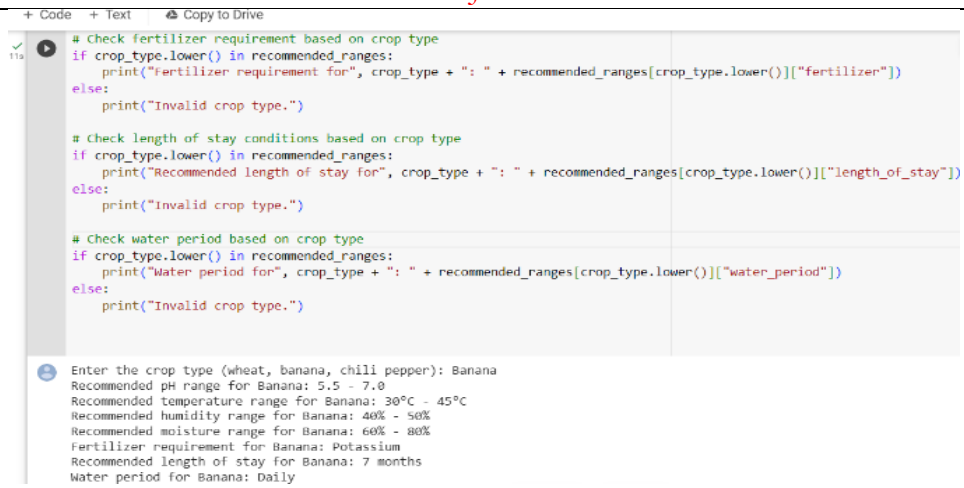

Figure 9: Crop-specific protocol for Banana crop

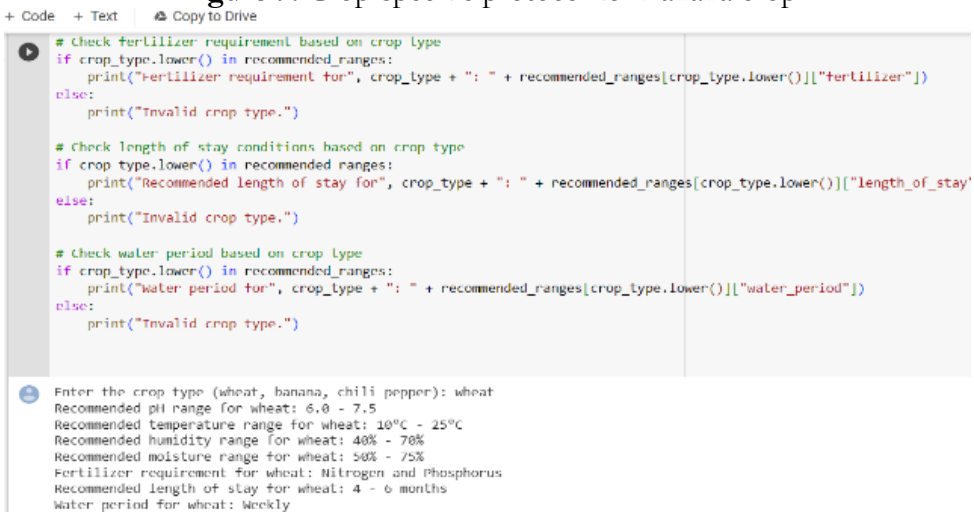

Figure 10: Crop-specific protocol for wheat crop

The Python script for Figure 7 provides the user interface to enter crop type (Wheat, chili, or banana) to get the crop recommendations. When the user enters the crop type such as Banana, the script provides the protocol specifically designed for the Banana crop as shown in Figure 9. The recommended ranges dictionary stores crop-specific recommendations such as pH range, Temperature range, Humidity range, Moisture range, fertilizer type, Length of stay, and Water period. It handles errors, for the crop type wheat, the script provides as shown in Figure 10:

- Soil pH range: 6 - 7.5
- Air Temperature range: 10°C - 25°C
- Humidity range: 40% - 70%
- Moisture range: 50% - 75%
- Fertilizer requirement: Nitrogen and Phosphorus
- Length of stay: 4 - 6 months
- Water period: Weekly

When the user enters any other crop other than mentioned above the script gives an error as "Invalid Crop type".

Research Findings and Conclusion:

This research presents an assessment of three prominent application layer protocols such as MQTT, AMQP, and HTTP. Initially, a general comparison and discussion of these application layer protocols introduces their characteristics. Subsequently, a more detailed analysis is conducted based on interconnected criteria to understand their strengths and

limitations. To facilitate this analysis, simple graphs are employed, offering a user-friendly perspective on each protocol's attributes relative to others. Then crop crop-specific protocol has been designed for three crops and basis of Python coding and confusion matrix the research questions have been addressed. Crop-specific algorithm also handles edge cases like extreme weather conditions or power outages that may disrupt communication in the field by deploying redundant sensors, integration of metrological data, Threshold alerts, and battery backup for power outages. In the Future, these results can be approved using scientific experiments and evaluation. The major contribution of this research is that MQTT is performing better for crop-specific IoT applications in agriculture, attributed to its low latency, high throughput, Quality of Service (QoS) capabilities for real-time monitoring, open standard nature ensuring compatibility, bandwidth efficiency in resource-constrained environments, and user-friendly implementation, supporting efficient data sharing and decision-making across diverse agricultural scenarios.

Utilizing crop-specific protocols in IoT applications for agriculture offers a range of tailored benefits that cater to the distinct requirements of each crop's cultivation. These specialized protocols enhance precision, efficiency, and yield by aligning technology with the unique characteristics of banana, wheat, and chili cultivation. By employing such protocols, farmers can ensure optimal growth conditions, targeted disease management, and timely interventions. For instance, in banana cultivation, a dedicated protocol can of different soil and environmental real-time parameters such as soil moisture, environmental temperature, humidity, soil pH, etc. It enables precise irrigation and prevents water-related stress. In wheat fields, crop-specific protocols aid in tracking growth stages, guiding farmers to apply fertilizers and treatments at critical junctures. In the context of chili crops, specialized protocols empower pest detection and control through timely data collection and analysis. Ultimately, embracing crop-specific protocols underscores a strategic approach to IoT integration in agriculture, resulting in improved resource allocation, minimized environmental impact, and maximized crop yield and quality for each unique cultivation context.

References:

[1] I. Has, M., Kreković, D., Kušek, M., & Podnar Žarko, "Efficient Data Management in Agricultural IoT: Compression, Security, and MQTT Protocol Analysis," *Sensors*, vol. 24, no. 11, p. 3517, 2024, doi: https://doi.org/10.3390/s24113517.

[2] E. A.-M. et Al, "Investigating Messaging Protocols for the Internet of Things (IoT)," *IEEE Access*, vol. 8, pp. 94880–94911, 2020, doi: 10.1109/ACCESS.2020.2993363.

[3] N. Naik, "Choice of effective messaging protocols for IoT systems: MQTT, CoAP, AMQP and HTTP," *2017 IEEE Int. Symp. Syst. Eng. ISSE 2017 - Proc.*, Oct. 2017, doi: 10.1109/SYSENG.2017.8088251.

[4] P. Singh, M. Kaur, and R. Bajaj, "An IoT-Enabled Crop Recommendation System Utilizing MQTT for Efficient Data Transmission to AI/ML Model," *Proc. Int. Conf. Circuit Power Comput. Technol. ICCPCT 2024*, pp. 315–320, 2024, doi: 10.1109/ICCPCT61902.2024.10672704.

[5] M. O. Al Enany, H. M. Harb, and G. Attiya, "A comparative analysis of MQTT and IoT application protocols," *ICEEM 2021 - 2nd IEEE Int. Conf. Electron. Eng.*, Jul. 2021, doi: 10.1109/ICEEM52022.2021.9480384.

[6] I. Gerodimos, A., Maglaras, L., Ferrag, M. A., Ayres, N., & Kantzavelou, "IoT: Communication protocols and security threats," *Internet Things Cyber-Physical Syst.*, vol. 3, pp. 1–13, 2023, doi: https://doi.org/10.1016/j.iotcps.2022.12.003.

[7] D. K. Bhoi, S. K., Ghugar, U., Dash, S., Nayak, R., & Bagal, "Exploring The Security Landscape: A Comprehensive Analysis Of Vulnerabilities, Challenges, And Findings In Internet Of Things (Iot) Application Layer Protocols," *Migr. Lett.*, vol. 21, no. s6, pp. 1326–1342, 2024, [Online]. Available: https://migrationletters.com/index.php/ml/article/view/8265

[8] M. B. Yassein, M. Q. Shatnawi, S. Aljwarneh, and R. Al-Hatmi, "Internet of Things: Survey and open issues of MQTT protocol," *Proc. - 2017 Int. Conf. Eng. MIS, ICEMIS 2017*, vol. 2018-January, pp. 1–6, Jul. 2017, doi: 10.1109/ICEMIS.2017.8273112.

[9] T. Moraes, B. Nogueira, V. Lira, and E. Tavares, "Performance comparison of iot communication protocols," *Conf. Proc. - IEEE Int. Conf. Syst. Man Cybern.*, vol. 2019-October, pp. 3249–3254, Oct. 2019, doi: 10.1109/SMC.2019.8914552.

[10] M. Diwan and M. D'Souza, "A Framework for Modeling and Verifying IoT Communication Protocols," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 10606 LNCS, pp. 266–280, 2017, doi: 10.1007/978-3-319-69483-2_16.

[11] A. N. Erdal ÖZDOĞAN, Osman Ayhan ERDEM and ÖZALP, "Adaptive Hybrid Application Protocol for IoT," *Acta Polytech. Hungarica*, vol. 21, no. 2, 2024, [Online]. Available: https://acta.uni-obuda.hu/Ozdogan_Erdem_Ozalp_142.pdf

[12] "(PDF) A SURVEY ON MQTT: A PROTOCOL OF INTERNET OF THINGS(IOT)." Accessed: Dec. 29, 2024. [Online]. Available: https://www.researchgate.net/publication/316018571_A_SURVEY_ON_MQTT_A_PROTOCOL_OF_INTERNET_OF_THINGSIOT

[13] S. Lakshminarayana, A. Praseed, and P. S. Thilagam, "Securing the IoT Application Layer from an MQTT Protocol Perspective: Challenges and Research Prospects," *IEEE Commun. Surv. Tutorials*, 2024, doi: 10.1109/COMST.2024.3372630.

[14] N. Nikolov, "Research of MQTT, CoAP, HTTP and XMPP IoT Communication protocols for Embedded Systems," *2020 29th Int. Sci. Conf. Electron. 2020 - Proc.*, Sep. 2020, doi: 10.1109/ET50336.2020.9238208.

[15] F. M. Khalid M. Hosny, Walaa M. El-Hady Samy, "Technologies, Protocols, and applications of Internet of Things in greenhouse Farming: A survey of recent advances," *Inf. Process. Agric.*, 2024, doi: https://doi.org/10.1016/j.inpa.2024.04.002.

[16] B. Wukkadada, K. Wankhede, R. Nambiar, and A. Nair, "Comparison with HTTP and MQTT in Internet of Things (IoT)," *Proc. Int. Conf. Inven. Res. Comput. Appl. ICIRCA 2018*, pp. 249–253, Dec. 2018, doi: 10.1109/ICIRCA.2018.8597401.

[17] R. Thakur, Rohit Kumar; Kumari, "A Comparison of Various IoT Application Layer Protocol," *Am. J. Electron. Commun.*, vol. 3, no. 1, pp. 28–34, 2022, doi: https://doi.org/10.15864/ajec.3106.

[18] C. Sharma and N. K. Gondhi, "Communication Protocol Stack for Constrained IoT Systems," *Proc. - 2018 3rd Int. Conf. Internet Things Smart Innov. Usages, IoT-SIU 2018*, Nov. 2018, doi: 10.1109/IOT-SIU.2018.8519904.

[19] N. Q. Uy and V. H. Nam, "A comparison of AMQP and MQTT protocols for Internet of Things," *Proc. - 2019 6th NAFOSTED Conf. Inf. Comput. Sci. NICS 2019*, pp. 292–297, Dec. 2019, doi: 10.1109/NICS48868.2019.9023812.

[20] P. Kiran and S. Shilpa, "Secure Communication Protocols for the IoT," *Secur. Commun. Internet Things Emerg. Technol. Challenges, Mitig.*, pp. 142–152, Jan. 2024, doi: 10.1201/9781003477327-12/SECURE-COMMUNICATION-PROTOCOLS-IOT-PRATHIBHA-KIRAN-SHILPA.

[21] A. Kondoro, I. Ben Dhaou, H. Tenhunen, and N. Mvungi, "Real time performance analysis of secure IoT protocols for microgrid communication," *Futur. Gener. Comput. Syst.*, vol. 116, pp. 1–12, Mar. 2021, doi: 10.1016/J.FUTURE.2020.09.031.

[22] H. ightiz, L., Yang, "A Comprehensive Review on IoT Protocols' Features in Smart Grid Communication," *Energies*, vol. 13, no. 11, p. 2762, 2020, doi: https://doi.org/10.3390/en13112762.

[23] I. Ben Hafaiedh, "Formal models for the verification, performance evaluation, and comparison of IoT communication protocols," *NCA 2022 - 2022 IEEE 21st Int. Symp.*

Netw. Comput. Appl., pp. 131–138, 2022, doi: 10.1109/NCA57778.2022.10013626.

[24] R. A. ALight, "Mosquitto: server and client implementation of the MQTT protocol," *J. Open Source Softw.*, vol. 2, no. 13, p. 265, 2017, doi: https://joss.theoj.org/papers/10.21105/joss.00265.

[25] P. Bhimani and G. Panchal, "Message Delivery Guarantee and Status Update of Clients Based on IoT-AMQP," *Lect. Notes Networks Syst.*, vol. 19, pp. 15–22, 2018, doi: 10.1007/978-981-10-5523-2_2.

[26] J. E. Luzuriaga, M. Perez, P. Boronat, J. C. Cano, C. Calafate, and P. Manzoni, "A comparative evaluation of AMQP and MQTT protocols over unstable and mobile networks," *2015 12th Annu. IEEE Consum. Commun. Netw. Conf. CCNC 2015*, pp. 931–936, Jul. 2015, doi: 10.1109/CCNC.2015.7158101.

[27] M. Singh, M. A. Rajan, V. L. Shivraj, and P. Balamuralidhar, "Secure MQTT for Internet of Things (IoT)," *Proc. - 2015 5th Int. Conf. Commun. Syst. Netw. Technol. CSNT 2015*, pp. 746–751, Sep. 2015, doi: 10.1109/CSNT.2015.16.

[28] "MPInspector: A Systematic and Automatic Approach for Evaluating the Security of IoT Messaging Protocols | Request PDF." Accessed: Dec. 29, 2024. [Online]. Available: https://www.researchgate.net/publication/362789914_MPInspector_A_Systematic_and_Automatic_Approach_for_Evaluating_the_Security_of_IoT_Messaging_Protocols

[29] M. H. R. Ronok Bhowmik, "An extended review of the application layer messaging protocol of the internet of things," *Bull. Electr. Eng. Informatics*, vol. 12, no. 5, pp. 3134–3141, 2023, doi: https://doi.org/10.11591/eei.v12i5.5236.

[30] B. B. L. Barlin O. Olivares, Julio Calero, Juan C. Rey, Deyanira Lobo and J. A. Gómez, "Correlation of banana productivity levels and soil morphological properties using regularized optimal scaling regression," *CATENA*, vol. 208, p. 105718, 2022, doi: https://doi.org/10.1016/j.catena.2021.105718.

[31] G. H. J. K. & J. A. S. R. A. Segura-Mena, J. J. Stoorvogel, F. García-Bastidas, M. Salacinas-Niez, "Evaluating the potential of soil management to reduce the effect of Fusarium oxysporum f. sp. cubense in banana (Musa AAA)," *Eur. J. Plant Pathol.*, vol. 160, pp. 441–455, 2021, doi: https://doi.org/10.1007/s10658-021-02255-2.

[32] H. Cui, Y. Luo, J. Chen, M. Jin, Y. Li, and Z. Wang, "Straw return strategies to improve soil properties and crop productivity in a winter wheat-summer maize cropping system," *Eur. J. Agron.*, vol. 133, p. 126436, Feb. 2022, doi: 10.1016/J.EJA.2021.126436.

[33] P. Malik, S. Sengupta, and J. S. Jadon, "Comparative Analysis of Soil Properties to Predict Fertility and Crop Yield using Machine Learning Algorithms," *Proc. Conflu. 2021 11th Int. Conf. Cloud Comput. Data Sci. Eng.*, pp. 1004–1007, Jan. 2021, doi: 10.1109/CONFLUENCE51648.2021.9377147.

[34] A. Khan, Muhammad Naeem, Rab, M. W. Khan, I. ud Din, and M. Khan, Muhammad Arif Khan, Muhammad Ayaz Ahmad, "Effect of zinc and boron on the growth and yield of chilli under the agro climatic condition of Swat," *Res. Artic.*, vol. 11, no. 3, pp. 835–842, 2022, doi: http://dx.doi.org/10.19045/bspab.2022.110084.