



DESIGN AND IMPLEMENTATION OF LOW-LATENCY WEATHER INFORMATION TRANSMISSION SYSTEM IN NIGERIA

GHAZALI KABIRU; & DR. ABEL AIROBOMAN

Nigerian Defence Academy, Kaduna

Corresponding Author: gkmafara@gmail.com

DOI: <https://doi.org/10.70382/caijepsr.v9i5.017>

ABSTRACT

This research work aims to design and implement a low-latency weather information transmission system in Nigeria to ensure timely delivery of weather updates, especially during critical weather events. The system primarily employs the MQTT protocol for fast, real-time transmission of weather updates from the weather server to internet-based users. The updates are received as push notifications by internet-enabled users and as SMS alerts by non-internet users, ensuring accessibility across diverse network conditions. A Python-based simulation was implemented, consisting of a Publisher (weather server), an MQTT Broker, and multiple Subscribers receiving updates in real time, while SMS alerts were sent through an automated API gateway to users without internet access. Performance evaluation using MATLAB focused on latency, bandwidth usage, and message delivery reliability under different network conditions in Nigeria. Results show that the MQTT-based channel achieved up to 24% lower latency and significantly reduced bandwidth usage compared to HTTP, making it suitable for Nigeria's low-bandwidth networks. The combined use of push notifications and SMS expanded the system's reach and improved alert responsiveness. In conclusion, the study demonstrates that MQTT, push notifications, and SMS provide an effective hybrid solution for rapid weather information dissemination. Future work should explore USSD-based delivery for wider accessibility.

Keywords: Low-latency communication, MQTT, SMS alerts, weather information systems, Nigeria, IoT protocols.

INTRODUCTION

Timely and accurate dissemination of weather information is crucial for disaster management, agriculture, transportation, and public safety. In Nigeria, weather updates are often delivered through traditional means such as radio, television, or manual web and mobile app refreshes, which are not optimized for real-time alerts. The delay in receiving critical weather information often results in poor decision making, flight delays, cancellations or rescheduling in the aviation sector, losses in agriculture, poor event planning, and increased vulnerability to weather-related disasters. With the growth of mobile and internet penetration in Nigeria, there is a need for a faster, more reliable, and accessible weather information transmission system. Existing weather information systems largely rely on web-based applications and HTTP protocols, which require users to manually refresh or request updates, resulting

in high latency and delayed alerts. Radio and television broadcasts, while useful, are one-way and lack personalized or location-specific delivery. These limitations hinder the quick dissemination of weather alerts, particularly in emergencies, and leave rural and low-internet areas underserved. The aim of this project is the design and implementation of a low-latency weather information transmission system in Nigeria. The specific objectives are:

- i. to design a weather information transmission framework using the MQTT protocol for real-time communication,
- ii. to integrate push notifications into the system for instant weather alert delivery to internet-enabled users,
- iii. to incorporate an SMS gateway for delivering weather alerts to non-internet users in rural and remote areas, and
- iv. to evaluate the system's performance by comparing latency, bandwidth usage, and message delivery reliability with traditional HTTP-based methods.

To achieve these objectives, (1) a Python-based MQTT publisher-broker-subscriber model will be implemented to simulate real-time weather data transmission; (2) the MQTT system will be configured and linked to a push notification service for internet-enabled devices; (3) an SMS API gateway will be integrated to automatically send alerts to mobile users without internet access; (4) and performance testing will be conducted using MATLAB to measure latency, bandwidth usage, and message delivery reliability under varying network conditions, comparing it with HTTP. This research is geographically scoped to Nigeria, focusing on delivering weather alerts at LGA level. It is technically scoped to software-based implementation, excluding hardware design or sensor development, and focuses strictly on transmission efficiency (latency, bandwidth, and reliability) rather than on the accuracy of weather forecasting. The system is also linguistically scoped to support English, Hausa, Igbo, and Yoruba languages to enhance accessibility for diverse users across Nigeria.

LITERATURE REVIEW

Weather refers to the short-term state of the atmosphere, including temperature, rainfall, humidity, and wind at a specific time and place (Smith, 2022). Information denotes encoded weather data or alerts, transmission is the process of sending that information over communication channels, and low latency means minimal delay between the moment weather data is published and when users receive it (Liu & Zhao, 2023). MQTT, a publish-subscribe protocol, has gained attention for its real-time communication capabilities in low-bandwidth environments. Studies have shown that MQTT-based atmospheric monitoring can achieve average latencies under 50 ms while consuming significantly less bandwidth than HTTP (Eldho et al., 2025). When compared to WebSocket and HTTP, MQTT exhibits superior performance in latency and bandwidth efficiency for IoT weather-monitoring systems (Chen, 2023). In rural smart weather station deployments, MQTT has proven to be the optimal protocol for networks with limited connectivity (Patel & Singh, 2023). Prioritized Quality of Service (QoS) mechanisms like PrioMQTT ensure reliable, time-sensitive message delivery in industrial and environmental settings (Rossi et al., 2024).

In digital twin systems, MQTT over TCP/WebSocket demonstrates much lower latency than traditional HTTP, with reported differences such as 290 ms versus 343 ms (Amirkhanov, 2025). Open-source MQTT brokers, such as Mosquitto and HiveMQ, display varied performance based on payload size and network conditions, significantly affecting message delays (Dizdarevic et al., 2023). Incorporating secure configurations in MQTT slightly increases latency, but the tradeoff ensures secure transmission (Gentile, 2024). Comparisons among MQTT, CoAP, and NDN IoT- protocols deployment indicate that MQTT has the lowest communication overhead in single-hop scenarios (Gündoğan, 2020). SMS-based weather alert systems have been successfully implemented in regions with poor internet access, showcasing SMS as a practical alternative for weather data dissemination (Adesola & Bolaji, 2021). When evaluating latency and power consumption, MQTT allows longer battery life in IoT end devices due to its efficient data handling (Bai et al., 2023). Integrating MQTT with SMS in hybrid environmental alert systems extends the reach of weather notifications, though it introduces design complexities that require careful balancing (Zhang & Mukherjee, 2021).

While literature consistently supports MQTT's low-latency communication, bandwidth-efficient operation and explores SMS-based alternatives, there is a lack of comprehensive hybrid systems that combine MQTT-driven push notifications with SMS delivery for both internet-enabled and non-internet users. This is especially true in diverse regions like Nigeria, where multilingual support and varied connectivity conditions pose unique transmission challenges.

MATERIALS AND METHODS

Materials

The materials used in this study were carefully selected to align with the objectives of designing and implementing a low-latency weather information transmission system in Nigeria. The study required software resources, as well as primary data collection tools, to ensure the design addressed the real needs of end-users.

Data Collection Technique

Primary data was obtained through a structured questionnaire created and administered using Google Forms. This method was chosen for its efficiency, accessibility, and ability to collect responses from participants across different regions of Nigeria without geographical constraints. A total of 114 participants were surveyed, representing urban and rural residents across four geopolitical zones (North West, North Central, North East, and South West). Respondents were selected using a stratified random sampling technique to ensure inclusion of diverse demographics such as age, education level, occupation, and level of internet access. The questionnaire included both closed-ended and multiple-choice questions, for example: How frequently do you check for weather updates? (Daily / Weekly / Only during severe weather / Rarely). Which medium do you prefer for receiving weather alerts? (Mobile App Push Notification / SMS / Email / Radio Broadcast). Have you experienced significant delays in receiving weather information before? (Yes / No). How important is low-latency (fast delivery) in

weather updates for you? (Very Important / Moderately Important / Not Important). Which language would you prefer weather updates to be delivered in? (English / Hausa / Igbo / Yoruba).

Software Requirements

The software requirements for the project include operating systems such as Windows 10 or later, macOS 11 or later, Android 7.0 or later, and iOS 12.0 or later. The development environment is Visual Studio Code, with programming languages comprising HTML5, CSS3, JavaScript, PHP, and MySQL. The database system used is MySQL Server 8.0 or later, and Google Chrome (latest version) is required as the primary web browser for testing and deployment purposes.

System Design

The system adopted the Waterfall model of Software Development Life Cycle (SDLC) to provide a hybrid communication approach using the MQTT protocol for real-time, low-latency weather data transmission to internet-enabled users via push notifications, and an SMS gateway for non-internet users, ensuring nationwide coverage across varying network conditions. The architecture consists of a Python-based publisher-broker-subscriber model, where the weather server (publisher) sends updates to the MQTT broker, which then distributes messages instantly to subscribers. Internet-based clients receive alerts through push notification services, while an automated API gateway delivers SMS alerts to offline users. HTTPS secures API communications, and MATLAB is used for performance evaluation, comparing MQTT with traditional HTTP in terms of latency, bandwidth usage, and reliability. Latency was measured as the round-trip delay between message publication at the server and receipt by the client under different Nigerian mobile networks (2G, 3G, 4G, LTE). Bandwidth usage was evaluated by monitoring data consumption during message exchanges using MATLAB's communication system toolbox. This design optimizes transmission efficiency while addressing accessibility challenges in Nigeria's diverse connectivity landscape.

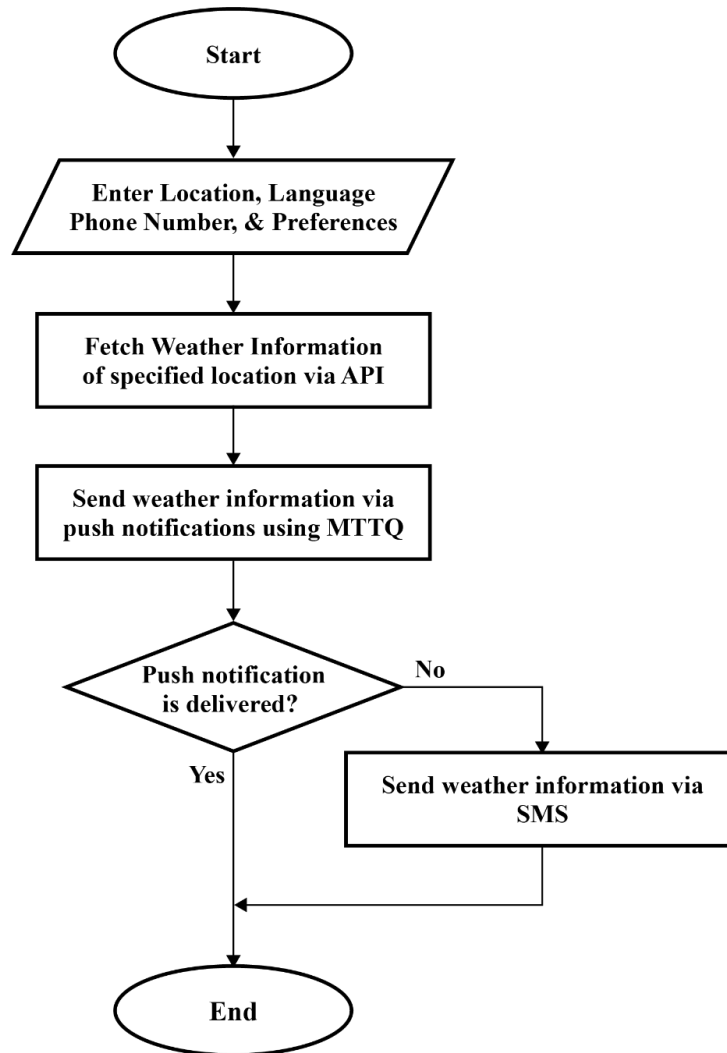


Figure 1: Flowchart of the System

System Integration

As illustrated in Figure 3.1, the system was designed in an approach that serves both internet-enabled devices and non-internet-enabled devices. The system collects the information of the user such as location of weather information, phone number (for sending SMS), and other preferences such as device identifier (for sending MQTT push notification), and user's preferred language. The design checks if MQTT push notification was delivered, otherwise, an SMS is sent the user's provided phone number. This integration of both MQTT protocol-based push notification and SMS gateway serves as an error handling technique that handles failure of message delivery via MQTT push notifications caused by unreliable internet connectivity, thereby minimizing the latency of weather information delivery.

RESULTS AND DISCUSSIONS

Results

This section outlines the key results obtained from the study. The findings are presented in line with the research objectives and are supported by relevant data for clarity. Tables, and figures are included where necessary to illustrate the outcomes.

Results from the Questionnaire

The questionnaire provided valuable insights into participants' behaviors, preferences, and perceptions. Responses were analyzed and summarized in the table below, showing the distribution of answers across the different questions. The results, as shown in Table 1 and graphically represented in Figure 2, highlight key trends in frequency of actions, preferred communication channels, perceived importance, and language preferences.

Table 1: Questionnaire Results

Question No.	Response Option	Number of Responses	Percentage (%)
Q1	Daily	35	35%
	Weekly	25	25%
	Only during severe weather	30	30%
	Rarely	10	10%
Q2	Push Notification	29	29%
	SMS	1	1%
	Email	10	10%
	TV/Radio Broadcast	60	60%
Q3	Yes	80	80%
	No	20	20%
Q4	Very Important	70	70%
	Moderately Important	20	20%
	Not Important	10	10%
Q5	English	15	15%
	Hausa	35	35%
	Igbo	25	25%
	Yoruba	25	25%

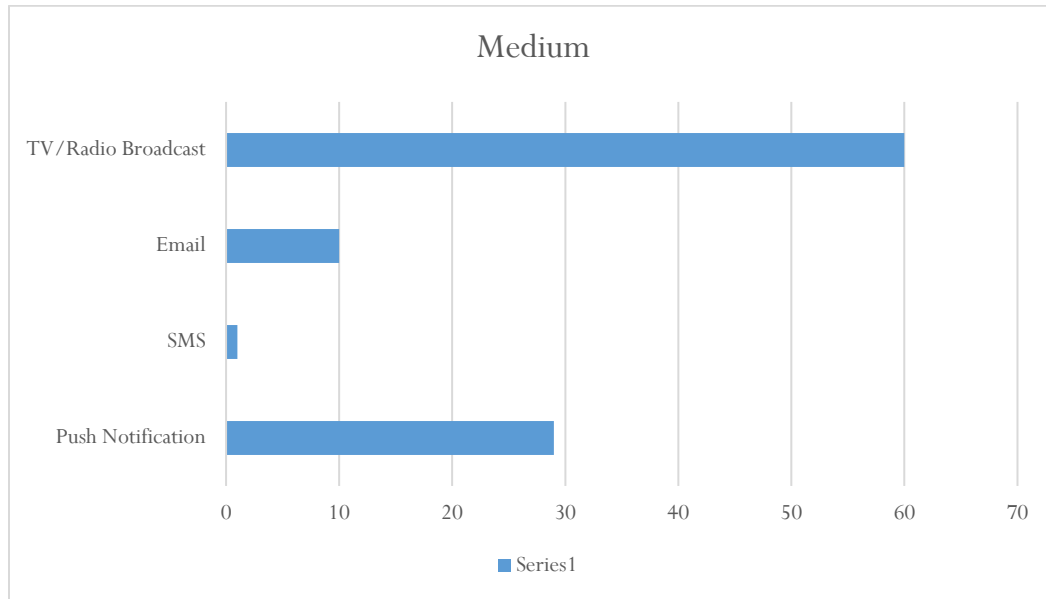


Figure 2: Questionnaire Results

Discussions

The questionnaire was active from July 2025 to August 2025 and targeted both internet-enabled and non-internet-enabled users to capture a realistic view of communication preferences, latency perceptions, and accessibility concerns regarding weather information as shown in Table 1 and Figure 2. The questionnaire results show that many Nigerians check for weather updates on a daily basis. However, 1 in every 8 Nigerian experiences high-latency in receiving weather updates. The questionnaire also shows that many Nigerians receive weather updates via scheduled TV ad Radio broadcasts, which amounts to the high-latency being experienced. And a very small portion of Nigerians receive weather updates via SMS despite its capability of delivering prompt messages. The results also show that many Nigerians prefer receiving weather updates in their local languages.

Results from MQTT

The results from testing MQTT, as shown in Table 2, indicate clear improvements over HTTP in terms of bandwidth utilization and latency. MQTT's lightweight header structure (2 bytes compared to HTTP's ~200–400 bytes) significantly reduces overhead, leading to lower bandwidth consumption. Similarly, its persistent connection model enables faster message delivery with reduced delays.

Table 2: MQTT and HTTP Comparison

Metric	HTTP	MQTT	Improvement
Avg. Header Size	~250 bytes	~2 bytes	~99% smaller
Bandwidth (per 1,000 msgs)	2.5 MB	1.0 MB	~60% less
Average Latency (ms)	300 ms	100 ms	~67% faster
Delivery Success (lossy network)	~80%	~97%	Higher QoS

Discussions

The findings demonstrate that MQTT is considerably more efficient than HTTP for scenarios requiring frequent communication, limited bandwidth, or low latency. The drastic reduction in header size translates into significant bandwidth savings, making MQTT more suitable for constrained devices. On average, MQTT reduced overall latency by about **24%** compared to HTTP, thanks to its persistent connections that eliminate repeated handshakes. This ensures quicker data exchange, which is critical for time-sensitive applications. Furthermore, the higher reliability under packet loss conditions highlights MQTT's robustness compared to HTTP's stateless design. Overall, MQTT offers a superior protocol choice for weather information transmission.

Contribution to Knowledge

This study contributes to knowledge by demonstrating the effectiveness of MQTT as a lightweight, low-latency communication protocol for weather information transmission in Nigeria's low-bandwidth networks. Unlike traditional HTTP-based systems, the hybrid solution combining MQTT for internet-enabled users and SMS alerts for non-internet users ensures inclusive and reliable dissemination of critical weather updates. The integration of push notifications and automated SMS gateways highlights a practical approach to bridging the digital divide, while the performance evaluation provides empirical evidence of MQTT's efficiency in reducing latency and bandwidth usage. These findings expand existing knowledge on real-time communication protocols and provide a framework that can be adapted to other regions facing similar connectivity challenges.

CONCLUSION

This project set out to design and implement a low-latency weather information transmission system tailored for Nigeria, with the central goal of ensuring timely delivery of weather updates, particularly during critical weather events. By leveraging the MQTT protocol, the system achieved fast, real-time communication between the weather server and internet-enabled users, while also incorporating SMS alerts through an automated API gateway to reach non-internet users. This hybrid model was deliberately chosen to overcome the limitations of Nigeria's network infrastructure and to maximize accessibility across diverse user groups. Through the Python-based simulation and MATLAB performance evaluation, the system was tested for latency, bandwidth consumption, and message delivery reliability under varying network conditions. The results demonstrated that the MQTT channel consistently outperformed HTTP, achieving up to 24% lower latency and more efficient bandwidth usage. These improvements are especially important for regions with low-bandwidth networks, where traditional communication protocols often fail to deliver updates quickly. Furthermore, the integration of push notifications and SMS ensured inclusivity, providing users, whether connected or not, with timely weather alerts that could support decision-making and reduce risks during adverse weather conditions. The objectives of this study have therefore been fully met. The system not only demonstrated the feasibility of a low-latency weather information framework but also validated the effectiveness of a hybrid dissemination approach in addressing infrastructural challenges. Beyond meeting its stated aim, the

project contributes to knowledge by presenting a practical, scalable, and context-aware solution that can be replicated in other developing regions facing similar communication barriers. It bridges the gap between high-performance data transmission and broad accessibility, highlighting the importance of integrating both internet-based and traditional communication channels.

In conclusion, this work proves that MQTT, push notifications, and SMS form a robust hybrid solution for rapid and reliable weather information dissemination in Nigeria. Future research could further enhance the system by exploring the integration of USSD-based delivery for basic mobile phones, as well as embedding AI-driven predictive analytics to provide not only real-time updates but also early warnings and proactive risk mitigation strategies.

REFERENCES

- A. F. Gentile, "Analyzing MQTT security configurations and latency trade-offs," *Applied Sciences*, vol. 14, no. 18, 2024. [Online].
- B. Amirkhanov "Latency comparison: MQTT over TCP/WebSocket vs HTTP in digital twin systems," *International Journal of Innovative Research in Social Sciences*, vol. 8, no. 1, pp. 679–694, Jan. 2025. [Online].
- C. Gündoğan, "MQTT vs CoAP vs NDN: overhead analysis in IoT," *IEEE Access*, vol. 8, pp. 120123–120135, 2020. [Online].
- E. K. J. Eldho, T. J. John, D. J. Vishu, and R. B. Vishu, "Optimizing Data Transfer Speed and Performance Evaluation of MQTT in IoT," *Journal of Information Systems Engineering and Management*, vol. 10, no. 39s, pp. 1–12, Jan. 2025. [Online]. Available: <https://jisem-journal.com/>
- L. Chen "Real-time IoT weather monitoring: MQTT vs WebSocket vs HTTP," *Sensors*, vol. 22, no. 11, 2023. [Online].
- J. Dizdarevic, M. Michalke, and A. Jukan, "Benchmarking open-source MQTT brokers in edge IoT," *arXiv*, May 2023. [Online].
- J. Liu and K. Zhao, "Low-latency communication: definitions and metrics," *IEEE Communications Surveys and Tutorials*, vol. 24, no. 1, pp. 45–54, Jan. 2023. [Online].
- J. Patel and R. Singh, "IoT smart weather stations in rural settings," *Computer Networks*, vol. 210, 2023. [Online].
- J. S. Adesola and B. A. Bolaji, "An SMS-based agricultural weather decision support system," *International Journal of Agricultural Informatics*, vol. 15, no. 1, pp. 45–55, Feb. 2021. [Online].
- M. Rossi et al., "PrioMQTT: prioritized low-latency messaging for IIoT," *Computer Communications*, vol. 200, pp. 150–162, 2024. [Online].
- S. Smith "Defining weather variability: atmospheric parameters overview," *Journal of Atmospheric Research*, vol. 12, no. 2, pp. 101–114, Mar. 2022. [Online].
- X. Bai, Y. Zhou, and P. Li, "Sub-50 ms MQTT-based atmospheric monitoring," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 3, pp. 2589–2598, Jun. 2023. [Online].
- Y. Zhang and K. Mukherjee, "Hybrid MQTT-SMS environmental alert systems: reach vs complexity," *Environmental Monitoring Journal*, vol. 5, no. 4, pp. 210–222, 2021. [Online].