6. 432-sari

by Jolene Hardwick

Submission date: 21-Jul-2025 03:17AM (UTC-0400)
Submission ID: 2691234263
File name: 6_432_Aisah.pdf (870.5K)
Word count: 3670
Character count: 21534

Infotmatics Management, Engineering, and Information System Journal, Vol. 3, No. 1, June 2025, 66-88
E ISSN : 3025-5384, DOI : https://doi.org/10.56447/imeisj

Smart Monitoring System Room Temperature Server Based on Iot Using Http Communication Protocol Method at Xyz University

Aisah Nur Endah Sari
STMIK Mardira Indonesia
Email: aisah.nurendahsari@stmik-mi.ac.id

Abstract

XYZ University, which relies on servers for administrative and academic functions, continues to employ manual temperature monitoring through physical site visits to the server locations. This method presents a risk of delayed response to temperature fluctuations, which could lead to operational issues and data loss. This project aims to establish a real-time temperature monitoring system for server rooms, utilizing IoT and employing the HTTP protocol for communication. The evaluation concentrates on assessing the latency of the HTTP protocol and the dimensions of data packets to verify the system's efficacy in real-time monitoring. The deployment of this intelligent monitoring system is expected to enhance the effectiveness of temperature regulation in server rooms and reduce the risk of damage from overheating. The research findings demonstrate that combining IoT with the HTTP protocol yields a more efficient, anticipatory, and sustainable temperature monitoring solution. Recommendations for future development include the integration of additional protocols, such as MQTT, CoAP (Constrained Application Protocol), or WebSocket.

Keywords : *Smart monitoring System Room Temperature, iot, http, telegram, Forward Engineering, ADDIE Model, dht11, ESP32*

INTRODUCTION

In this rapidly evolving information technology era, numerous organizations rely on computerized systems to conduct their activities. A vital element of these systems is the server, which functions as the primary repository for data and information processing. The 2018 draft of the Minister of Communication and Information Technology Regulation stipulates that server room temperatures must be maintained between 18 °C and 24°C to guarantee optimal server functionality. Furthermore, data center providers, as indicated in Article 2, paragraph (3), are accountable for maintaining the continuity of data center services customized to their operational requirements. One of these roles involves implementing an environmental monitoring system that includes temperature and humidity surveillance in the data center.

XYZ University manages its servers for administrative and academic functions, necessitating the maintenance of appropriate temperature settings to avert performance disruptions and reduce the risk of overheating-related damage. Nonetheless, the temperature surveillance of the server room remains a manual process, necessitating personnel to visit the site in person. This monitoring relies on the supervisor's recollection, which may result in insufficient oversight if forgotten, frequently leading to a lack of reactivity to sudden temperature fluctuations. Procrastination in resolving these issues can significantly impact campus operations, leading to network disruptions and the loss of critical data.

Consequently, there is an immediate necessity to establish an intelligent monitoring system capable of real-time temperature surveillance.

The implementation of Internet of Things (IoT) technology is a viable solution to this problem. The societal and personal advantages of IoT are achieved through M2M (Machine-to-Machine) advancements. In addition to facilitating internet access, IoT enhances efficiency through work automation, information exchange, updates, modifications, threshold maintenance, and variation comparison. Machines will engage in direct communication through intelligent algorithms, liberating us from mundane chores, raising the quality of life for end users, diminishing complexity and cycle durations, augmenting efficiency, and frequently elevating safety (VSSS Kalaga Rao, 2022). The Internet of Things (IoT) is being increasingly adopted across various sectors to address challenges that require automated monitoring and control, as exemplified by temperature monitoring systems in server rooms. Smith et al. (2018) found that employing IoT for room temperature monitoring can enhance operational efficiency, facilitate more accurate oversight, and enable prompt responses to temperature fluctuations (Pratama, 2022).

The HTTP protocol has been selected as the principal communication technique to enable interaction among devices within this IoT system. The evolution of the HTTP/HTTPS protocol remains susceptible to further enhancement. A significant opportunity exists in developing more efficient and secure encryption methods, which can reduce the computational load on both servers and clients. Moreover, the integration of HTTP/HTTPS with advanced technologies, such as IoT, presents an opportunity to create more complex and secure web applications (Shafira Nur Alfiah, 2024). Nonetheless, given that HTTP was initially developed for a broad internet connection, it is essential to guarantee that it can deliver suitable communication latency for monitoring requirements.

Evaluating the latency of HTTP in this system is essential to guarantee that end users can function efficiently with dependable performance in network-based applications (Hanif & Amri, 2023). This research aims to evaluate the efficacy of the HTTP protocol in meeting the requirements of a real-time temperature monitoring system by examining HTTP latency and the dimensions of the data packets transmitted.

Given the aforementioned conditions, the deployment of an IoT-based intelligent monitoring system utilizing the HTTP protocol, corroborated by the outcomes of this testing, is anticipated to facilitate effective, efficient, and proactive temperature regulation in the server room at XYZ University, thereby ensuring temperature stability and safeguarding server equipment from potential damage.

METHOD

This study employs a forward engineering methodology and implements the system using the ADDIE paradigm, utilizing HTTP as the communication protocol with the POST method (Mesra, 2023). The subsequent steps are as follows:

This study employs a forward engineering methodology and implements the system using the ADDIE paradigm, utilizing HTTP as the

communication protocol with the POST method (Mesra, 2023). The subsequent steps are as follows:

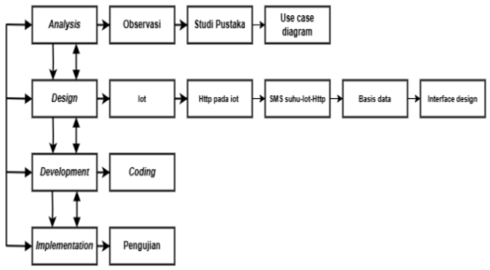

Figure 1. Research Stages

Analysis: This phase involves a comprehensive examination of the material into its constituent elements to identify and assess prevailing concerns, facilitating suggested enhancements. This procedure entails evaluating the system requirements and the requisite hardware for development. Data is gathered through observations, interviews, and literature research and represented using use case diagrams.

Ongoing Business Process

Following the completion of observations and interviews, the existing activity process is delineated as follows:

1) Staff recalls whether to watch the server room.
   a. If the personnel fails to recall, no oversight is conducted, and the procedure concludes.
   b. The process proceeds if the staff recalls.
2) Should the personnel recall, they will proceed directly to the server room for monitoring.
3) Upon arrival in the server room, the team verifies the normalcy of temperature and humidity levels.
   - If the temperature and humidity are within typical ranges, no additional measures are required, and the process concludes.
   - If the temperature and humidity are abnormal (for instance, an increase in temperature resulting in a hot environment indicates damage), the process persists.
4) If the condition is abnormal, the personnel will attempt to rectify the issue by modifying the AC temperature.
5) Following the adjustment of the AC temperature, the personnel verifies if the issue has been rectified.
   - The process concludes upon resolution of the issue.
   - If the issue remains unresolved, the procedure persists.

6) Should the staff be unable to resolve the issue alone, they will contact a technician. The technician will thereafter address the issue.
7) Following the technician's intervention, a verification is conducted to ascertain whether the issue has been resolved.

- The process concludes upon resolution of the issue.
- If the issue remains unresolved, the process reverts to the technician call step (loop).
- The following is illustrated with a flowchart:

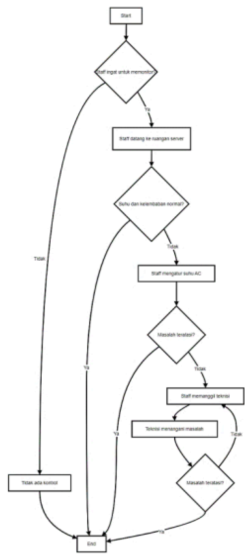

Figure 2. Ongoing Business Process

Needs Analysis

Analysis of User Requirements

Requirements for users in the design process of IoT-based systems, encompassing:

1) IoT Communication Anytime, Any Context • The system is accessible and operable around the clock.
    - Applicable in diverse contexts and scenarios

- Flexibility in usage time • Compatible with any device • Applicable to a variety of device types
- Facilitates multi-device integration • Operates in any location • Extensive coverage without geographical limitations • Applicable in diverse settings • Enables mobility and freedom in location choice

Any Route Any Network (Any Route & Network)

2) Accommodates multiple network protocols. The chosen network protocol is the HTTP communication protocol.
3) Development of an intelligent surveillance system.
4) Evaluation of the HTTP communication protocol.

Proposed Smart System Analysis With Use Case Diagram

A Use Case diagram illustrates the collaborative tasks of Admin 1 and Admin 2 in managing the online system and Telegram notifications. The proposed use case diagram for Admin 1 and Admin 2 is as follows:

Figure 3. Proposed Smart System With Use Case Diagram

The following is a description of the Use Case diagram for admin 1 and admin 2:

Table 1. Use Case Diagram Description

No.	Use Case	Description
1.	Login	Access rights for system users
3.	Monitoring Dashboard	Admin can monitor the dashboard, which contains server room information including temperature and humidity in real time
3.	Manage Server Room Data	Admin adds, edits, deletes room server data, including Room Server Name, Iot Devie Code, Set notification Upper limit, Set notification lower limit.
4.	History Monitoring	Admin can monitor the temperature and humidity history in the server room, there is a device code, status that provides

		information on the condition of normal temperature and humidity or whether it is increasing or decreasing in real-time.
5.	Manage Device Notification Settings	Admin can create, edit, delete and manage client id app that contains Telegram username, Telegram phone, Telegram chat id. The goal is to provide temperature and humidity notifications via telegram that have been inputted in the Setting Device Notification feature and can be to several users.
6.	Manage Users	Admin can create, edit, and manage the users.
7.	Manage App Config	Admin can manage app data. There are app name, activation code, app description, app keyword, app author.
8.	Manage My Account	Admin can update account data.
9.	Monitoring Normal Temperature and Humidity Values Telegram Notifications Every 30 Minutes	Admin can monitor telegram notifications containing messages about normal temperature and humidity every 30 minutes.
10.	Reading Telegram Notifications Abnormal Temperature and Humidity	Admin can read the telegram notification sent by the alert system if before 30 minutes there is a high or decreasing temperature and humidity, then the notification will appear at that minute or second..

2. Design and Modelling. This stage, informed by the analytical results, emphasizes modeling and design, specifically concentrating on the development of IoT systems, the application of HTTP in IoT, the creation of an IoT-based innovative temperature system utilizing the HTTP communication protocol, as well as interface and database design. The IoT tool's design utilizes https://wokwi.com. The subsequent are the design outcomes:

IoT Architecture On Smart Monitoring System

This is an IoT architecture, as described by the author, devised to enhance IoT and M2M (machine-to-machine) communication, tailored to the requirements of the XYZ University case study.

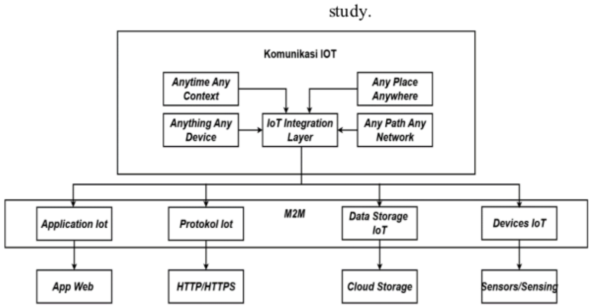

Figure 4. IoT Architecture

1. At any moment, The IoT-enabled intelligent monitoring system is perpetually accessible and operational, offering flexibility for use at any time.

2. The IoT in this intelligent monitoring system accommodates a diverse array of devices.
3. Any Location at Any Time
   - Accessibility from multiple sites. Accessible from any location, unrestricted by distance.
   - Any Route Any Network
   - Adaptability in network selection.
   - M2M

a) Internet of Things Devices:

Sensors: To get data from the surrounding environment.

This intelligent monitoring system employs IoT devices equipped with sensors, specifically DHT11, ESP32, and an LCD.

a) Internet of Things Data Storage

Data measured by the DHT11 is transmitted to the ESP32, where it is subsequently stored in cloud storage, a type of data storage on a cloud server.

b) Internet of Things Protocol

The communication mechanism utilized by this intelligent monitoring system employs HTTP/HTTPS.

c) Internet of Things Application

The utilized intelligent monitoring system application is web-based.

HTTP Architecture In Smart Monitoring System

Below is the Http protocol architecture in the Smart Monitoring System with the following explanation:

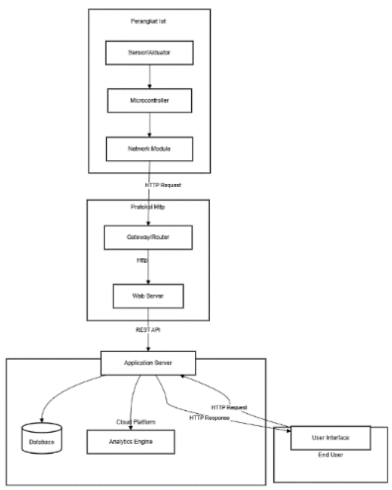

Figure 5. HTTP Architecture on IoT

1. IoT Device Layer: • Sensors quantify physical parameters, including temperature (DHT11) and humidity, among others.

- Microcontroller: Process raw sensor input, execute analog to digital conversion (ADC), and operate firmware for fundamental control logic.

In the author's investigation employing a microcontroller, ESP32

- Format data appropriately (e.g., JSON, XML)
- Network Module: Manage internet connectivity (WiFi, Ethernet, GSM).

2. HTTP Protocol Layer: • Gateway/Router: Facilitates the connection of IoT devices to the internet • Web Server: Manages HTTP requests and responses
3. Cloud Platform Layer: • Database: Retains IoT data • Analytics Engine: Conducts real-time data processing, batch processing for historical data, and prepares data visualization.
4. End-User Layer: User Interface - Interface for oversight and regulation. Information transmission:
   a. Data Collection: • Sensors acquire physical data • Microcontroller Unit (MCU) processes unrefined data • Data is structured in JSON/XML format • Network module composes HTTP request
   b. Data Transmission: • The device initiates an HTTP POST request. • Data is encrypted by utilizing HTTPS. • The request traverses a gateway for NAT and routing. • The web server receives and validates the request. c. Data Processing: • The application server authenticates the device and data. • Data undergoes validation. • Business rules are implemented. • Data is stored in the database. • Real-time analytics are conducted.
   c. Data Consumption: • The user interface requests data using the REST API. • The application server retrieves and formats the data. • The data is transmitted to the user interface. • The interface presents the data visually.

Smart System Architecture Room Temperature Server Based on IoT Using HTTP Communication Protocol Method

Here is the architecture:

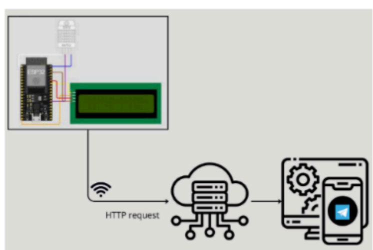

Figure 6. Architecture of the IoT-Based Server Room Temperature Smart System Using the HTTP Communication Protocol Method

This document outlines the design of the innovative system's room temperature server, which operates based on IoT principles and utilizes the HTTP communication protocol.

The DHT11 sensor quantifies the room's temperature and humidity, transmitting this data to the ESP32 microcontroller, which then displays the information on the LCD screen.

Subsequently, the ESP32 transmits the data to the cloud server using the HTTP communication protocol, employing Wi-Fi as the channel that connects the IoT device to the internet, the data storage server, and other systems.

The data is stored in the database and then processed.

4) Verifying if the temperature or humidity surpasses the designated threshold (abnormal) or remains within normal parameters.
5) If the condition surpasses the threshold (abnormal), the system will display a status in the historical monitoring menu. If the temperature exceeds the upper limit, the server transmits a message to the Telegram Bot indicating that the temperature and humidity are anomalous for that specific minute and second.
6) Under typical temperature and humidity conditions, the system will display a status in the historical monitoring menu indicating normal temperature and humidity levels. The server transmits messages to the Telegram bot systematically every 30 minutes.
7) Notifications may be received on several devices.
8) The server updates the online dashboard in real time, both under normal conditions and during an alert. Users can directly monitor data via the web application.

Develop: During this phase, product development progresses according to the previously established specifications, encompassing the coding process using the C++ programming language and Arduino software.

Implement: Prior to implementation, system testing is performed via black box testing methodologies. Upon successful completion of testing, the product is deployed and utilized via the HTTP communication protocol.

RESULT AND DISCUSSION

Upon completion of all processes, the next step is to examine the results and analysis of this system to verify compliance with the established standards. Testing is performed first. Upon successful completion of the tests, the system is subsequently deployed.

A. Quality of Service (QoS) testing

In the HTTP performance evaluation utilizing Quality of Service (QoS) analysis with the Wireshark application.

a) HTTP Performance Evaluation (Latency/Response Time)
During the HTTP performance evaluation. This study involves data collection from a DHT11 sensor and web-based monitoring, with latency assessed using the Wireshark application for 30 minutes. The results are exported as a CSV file for latency calculation in Excel. The equation for determining delay:

$$\text{Average } Delay = \frac{Total\ Delay}{\text{Package Received}}$$

Results = 416,000 ms
Total Delay = 1794,529423 s
Average Delay = 0,416557433 s
Make it into ms = s x 1000
= 416,000 ms

The latency classification is moderate.

b) HTTP Performance Evaluation (Packet Loss)
Packet loss is a metric that signifies the occurrence of packets that fail to transmit

successfully, often owing to collisions and network congestion. Below is how to compute the packet loss value:

$$Packet\ Loss = \frac{(\text{Data Packet sent} - \text{data packet received})}{100\%\ \text{Data packet sent}}$$

Or by using the filter method in the Wireshark application with:

tcp.analysis.ack_lost_segment

Results = 0%

So there is no packet loss during delivery.

c) Http Protocol Performance Testing (Throughput)

Throughput is the efficient data transfer rate quantified in bits per second (bps). Throughput is defined as the total number of packet arrivals recorded at the destination within a specified time interval, divided by the length of that interval. The formula is as follows:

$$Throughput = \frac{(\text{Data packet received})}{\text{Observation Period}}$$

Number of Bytes : Time span, s =

2024814 : 1794.529 = 1,12832615 bytes/s

Change the value to bytes x8

1,12832615 bytes/s x 8 = 9,026 kbits/s

The throughput value obtained is, 9,026 kbits/s

B. Tool Testing

The table below indicates that the tool performed as anticipated throughout testing.

Table 2. Tool Testing

No.	DHT11 Sensor		LCD	Telegram
	Temperature	Humidity		
1.	<28	>50	COME ON STAGE	Terkirim
2.	27	64	COME ON STAGE	Terkirim
4.	26	65	COME ON STAGE	Terkirim
5.	24	75	COME ON STAGE	Terkirim

C. Software Testing

The software demonstration was performed to verify that the final product operates as intended.

1. Results of Web Application Testing

The web application's ability to collect data from multiple DHT11-ESP32-cloud sensor setups met expectations. The data can be observed in real-time on the web monitoring dashboard. This is illustrated in the graphic below:

Figure 7. Monitoring Dashboard

2. Outcomes of Testing Routine Telegram Notifications at 30-Minute Intervals

The test results indicate that the temperature and humidity can be automatically transmitted by the Telegram bot titled "IOT NOTIFICATION" at the designated time. Notifications will be dispatched regularly every 30 minutes if the temperature and humidity are within typical ranges. The subsequent content is an image:

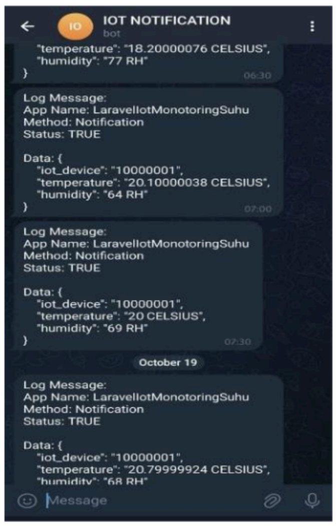

Figure 8. Routine Telegram Notifications every 30 minutes

3. Results of the Telegram Unlimited Notification Test

The test findings are timeless, so any abnormal fluctuation in temperature or humidity within 30 minutes would trigger an immediate notification within the exact second or minute. The image below depicts the limitless notification that was tested and fulfilled expectations.

Figure 9. Telegram Notifications Unlimited Time

Prior to the 19th, the temperature was standard between 10:30 and 11:00 AM, with messages dispatched every 30 minutes. On the 20th, at 8:07 AM, shortly after the gadget had been functioning for under 30 minutes, the temperature rose, prompting an urgent automatic notice.

CONCLUSION

From the diverse reasons provided, multiple conclusions can be drawn as follows:

1. Effectively created an intelligent monitoring system. This system's intelligence is in its ability to deliver early notifications and alerts to personnel when the temperature deviates from the established standard through Telegram notifications. Regardless of the normal temperature, Telegram notifications are consistently dispatched every 30 minutes.
2. Effectively developed an IoT architecture for the intelligent monitoring system, using an HTTP framework for its implementation, hence enhancing IoT and M2M interactions. The Internet of Things (IoT) operates in any time, context, device, location, and network, encompassing machine-to-machine (M2M) devices, data storage, protocols, and applications.
3. The HTTP communication protocol was effectively implemented on an IoT-based server room temperature innovative system and subjected to QoS analysis via the Wireshark application, yielding a latency of 416,000 ms, categorized as moderate, a packet loss of 0%, indicating no data loss and classified as very good, and a throughput of 9,026 kbits/s.

REFERENCES

Cirani, S., Ferrari, G., Picone, M., & Veltri, L. (2019). Internet of Things: Architectures, protocols and standards.

Diyasa, I. G. S. M., Winardi, S., Idhom, M., & Budianto, A. (2024). Pembelajaran Internet of Things dengan modul trainer ESP-32. Thalibul Ilmi Publishing & Education. https://books.google.co.id/books?id=gyDwEAAAQBAJ

Kominfo RI. (2018). RPM Data Center 2018 (Vol. 13, pp. 1689–1699).

https://web.kominfo.go.id/sites/default/files/users/3997/RPM%20PUSAT%20DATA%20STANDAR%20KP120118.pdf

Mauluddin, A., Setiawan, A., & Supriadi, I. (2022). Prototype single IP login dan kriptografi asimetris digital signature algorithm pada user authentication. J. Tiarsie, 19, 37–42. https://jurnalunla.web.id/tiarsie/index.php/tiarsie/article/view/134

Mesra, R. (2023). Research & development dalam pendidikan.

Nasir, M., Saputri, N., & Dwina, N. (2024). Internet of Things: Aplikasi dan penerapan mikrokontroler Arduino. Penerbit Andi. https://books.google.co.id/books?id=KZUNEQAAQBAJ

Pratama, M. R. (2022). Implementasi IoT untuk monitoring suhu ruangan dengan ESP32. Jurnal XYZ, 2, 1–19.

Reza, Z. (2021). Sistem monitoring jaringan berbasis web dengan notifikasi Telegram. Jurnal XYZ, 4, 4–21.

Shafira Nur Alfiah, A. A. H. (2024). Memahami http/https: Perspektif baru dan peluang inovasi. Jurnal XYZ, 1, 81–84.

6. 432-sari

ORIGINALITY REPORT

1%	1%	0%	1%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to University of Melbourne Student Paper	<1%
2	jurnal.stmik-mi.ac.id Internet Source	<1%
3	www.kluniversity.in Internet Source	<1%
4	www.ecoi.net Internet Source	<1%

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		