INTERNATIONAL JOURNAL OF ELECTRONICS AND COMPUTER APPLICATIONS

ORIGINAL ARTICLE

Article access online

OPEN ACCESS

Received: 12.01.2025

Accepted: 09.06.2025

Published: 30.06.2025

Citation: Patil V, Ingale G, Gaikwad A, More AP. (2025). Energy Efficient Routing Protocol for IoT-enabled MANETs. International Journal of Electronics and Computer Applications. 2(1): 1-8. https://doi.org/10.70968/ijeaca.v2i1.C1001

* **Corresponding author**.

aparna.more@moderncoe.edu.in

Funding: None

Competing Interests: None

ISSN
Print: XXXX-XXXX
Electronic: 3048-8257

Energy Efficient Routing Protocol for IoT- enabled MANETs

Vaishnavi Patil[1], Gayatri Ingale[1], Aishwarya Gaikwad[1], Aparna P More[2]*

1 Department of Electronics and Computer, PES Modern College of Engineering, Pune (Affiliated to Pune University), Pune, Maharashtra, India
2 Assistant Professor, Department of Electronics and Telecommunication, PES Modern College of Engineering, Pune (Affiliated to Pune University), Pune, Maharashtra, India

Abstract

The rapid expansion of Internet of Things (IoT) devices has facilitated the development of Mobile Ad Hoc Networks (MANETs), where wireless nodes interact without relying on any centralized infrastructure. Nevertheless, the highly dynamic topology of MANETs and the limited battery capacity of IoT devices create major obstacles for energy-efficient routing. To address this, this paper introduces an Energy Efficient Routing Protocol (EERP) specifically designed for IoT-based MANETs. The proposed EERP integrates principles of both distance-vector and link-state routing to balance energy usage and sustain network performance. By incorporating local energy information and adaptable routing mechanisms, the protocol effectively reduces energy drain and extends the operational life of network nodes.

Keywords: Internet of Things; Sensors; MANETs; Routing Protocol; Duty Cycle; TEEN

Introduction

The rapid proliferation of Internet of Things (IoT) devices has revolutionized communication paradigms, enabling a wide array of applications in smart cities, healthcare, and industrial automation. Mobile Ad Hoc Networks (MANETs), characterized by their decentralized structure and dynamic topology, provide an ideal framework for such IoT applications, allowing devices to communicate seamlessly without the need for fixed infrastructure. However, the unique challenges presented by MANETs, particularly in terms of energy consumption, pose significant barriers to their effectiveness and sustainability. IoT devices are often constrained by limited battery life and energy resources, making energy efficiency a critical concern in the design of routing protocols. Traditional routing protocols typically prioritize metrics such as throughput and latency, often neglecting the energy aspect. This oversight can lead to rapid depletion of battery resources, resulting in network instability and reduced connectivity. As the network scales and node density increases, these challenges become even more pronounced. Existing routing protocols for MANETs often prioritize metrics such as throughput and latency, neglecting energy consumption. This oversight can lead to rapid battery depletion and network instability as nodes become unavailable, resulting in increased packet loss and latency. To address these challenges, there is a growing need for energy-aware routing protocols that balance energy efficiency with network performance.

Such protocols can enhance node longevity and improve overall network sustainability. The development of an Energy Efficient Routing Protocol (EERP) specifically designed for IoT-enabled MANETs aims to optimize energy consumption while maintaining reliable communication, addressing the unique constraints and requirements of this evolving technological landscape. To address these issues, we proposed an idea of an Energy Efficient Routing Protocol (EERP) specifically designed for IoT-enabled MANETs. EERP aims to balance the trade-offs between energy consumption, data delivery reliability and network performance. By integrating adaptive routing strategies that prioritize energy-efficient paths and local energy metrics, our protocol seeks to enhance the longevity of nodes and ensure sustainable operation in dynamic environments. This introduction lays the groundwork for examining the design and assessment of EERP, emphasizing its promise to enhance both the efficiency and longevity of IoT-integrated MANETs in practical scenarios.

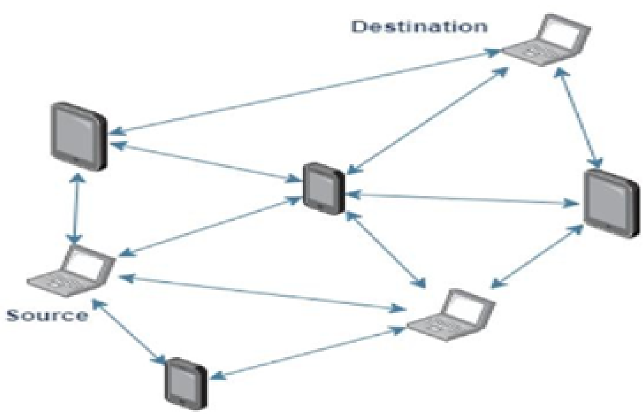

Fig 1. Mobile Ad hoc Network (MANET)

Literature Review

This system employs MANET technology using the IoT sensor nodes, to receive the data from client side and wirelessly send it using the LoRa module to the sensor side. Energy- efficient routing protocols are also implemented, that aim to balance the trade-offs between energy consumption, network performance, and data delivery reliability.

In this paper, V. Anjana Devi et al. introduced a secure protocol for MANET-IoT networks aimed at enhancing energy-efficient routing and defending against malicious network attacks. Their approach includes mechanisms like the Energy Efficient Routing Establishment (EERE) scheme and the Aggregated Packet Control Trust Protocol (APCTP), which collaboratively select optimal routes to ensure secure data transfer within MANET-IoT systems. Additionally, the Whale Positions Integrated Flower Pollination Algorithm (WP-FPA) is utilized to further refine energy-efficient path selection. The APCTP protocol also plays a key role in detecting severe attacks such as denial of service (DoS), collisions, jamming, and black hole threats, which can significantly undermine network performance and lifespan. [1]

Fig 2. Internet of Things (IoT)

In this paper, Quy Khanh Vu and Ngoc Anh Le introduced an energy- aware routing protocol for MANETs, named EED, which employs a load distribution strategy guided by a cost function that considers both the hop count and the residual energy of nodes to choose the most suitable path. Simulation outcomes demonstrated that EED outperformed conventional protocols such as DSR and AODV in terms of energy consumption, network longevity, and packet delivery performance. Effective routing protocols for MANETs should account for route length as well as link quality and stability. With the goal of enhancing energy efficiency, this study emphasizes identifying routing factors that significantly influence network lifespan and presents an energy-optimized protocol, derived from the Dynamic Source Routing (DSR) protocol, referred to as EED (Energy Efficient-DSR). [2]

In this paper, Hari Varshini S et al. explored a fire alarm system solution aimed at mitigating the severe risks associated with fire incidents. Developing a smart IoT-based fire alarm system requires integrating various hardware components, including the ESP8266 NodeMCU, flame sensor, and buzzer, all connected to a computer for data processing and communication. Additionally, the system employs the Blynk application to transmit alerts to the user. The proposed fire alarm setup combines both software and hardware modules, enabling rapid detection and response to fire hazards. This framework is suitable for deployment in environments such as chemical plants, shopping centers, local stores, educational institutions, parking facilities, and corporate offices. [3]

In this paper, Munisha Devi and Nasib Singh Gill studied the overview of MANETs and its routing protocols by highlighting their functionality, benefits, characteristics and limitations. MANETs and IoT-enabled smart environment is also discussed here. The introduction of MANET in smart environment needs new protocols for connecting devices to the internet. Comparative study of various protocols has been also done. It is difficult to achieve power and security awareness across these networks due to dynamically changing topology and infrastructure less features. Therefore, the power awareness mechanism and security facilities for all kinds of applications in ad hoc network should be built. [4]

In this paper, Latha Maheshwari et al. presented a project aimed at detecting forest fires and notifying forest officers promptly when such incidents occur. Studies indicate that around 85% of forest fires are caused by human actions, while the rest result from natural factors like lightning. The system uses an advanced IoT processor to manage its operations, along with multiple sensors—including temperature, smoke, and flame sensors—to detect fire outbreaks. Once a fire is identified, the system pinpoints its exact location and communicates this information to the nearest forest officer using a GSM module and Bluetooth module. When abnormal conditions are sensed, the Arduino executes the programmed instructions, and the GSM module sends alerts to designated personnel. Furthermore, the system is entirely IoT-enabled, with continuous monitoring and data logging on online platforms, allowing officials to access real- time updates and review stored data whenever needed. [5]

In this paper, Sania Khan et al. developed a system designed to detect and mitigate disasters caused by forest fires. A key advantage of this forest fire detection system is its rapid response, which enables forest officials to receive early warnings and take swift action to contain the fire before it spreads extensively. By transmitting critical information promptly, the system enhances early detection, thereby reducing environmental and ecological damage. The detection mechanism is highly accurate, with minimal chances of false SMS alerts. When a fire is identified, a location URL is sent immediately, allowing quick dispatch of assistance to the affected site, which helps control the blaze and limit its spread. Real-time sharing of disaster data further reduces risks to human life and protects wildlife habitats. [6]

Methodology

This system employs MANET technology using the IoT sensor nodes, to receive the data from client side and wirelessly send it using the LoRa module to the sensor side. Energy- efficient routing protocols are also implemented, that aim to balance the trade-offs between energy consumption, network performance, and data delivery reliability.

Let us summarize the key components and their functionalities:

1.Client Device (ESP32 with LoRa Module, GPS, and Sensors):

- ESP32: Core processing unit.
- DHT11 Sensor: Measure humidity and temperature.
- KY-026 Flame Sensor: Detects the presence of flame.
- MQ-9 Gas Sensor: Detects smoke and gas concentrations.
- NEO-6M GPS Module: Captures real-time locations.
- RA-02 LoRa Module: Provides long range communication between client and server.

2. Server (ESP32 with LoRa Module and GSM):

- ESP32 with LoRa Module listens for data from Client Node.
- SIM800L GSM Module: Semds SMS Alerts to fire
- Control authorities in case of fire detection.
- The sensor values are displayed on online platform such as ThingSpeak, while the energy comparison graphs and data is displayed on NetAnim and XGraph using NS3.

ThingSpeak: Online Platform

ThingSpeak is a cloud-driven IoT platform designed to gather, store, analyze, and visualize data from networked sensors and devices. It serves as an intermediary connecting physical hardware—such as Arduino, Raspberry Pi, ESP8266, ESP32, or various industrial sensors—to cloud-based data processing and analysis tools. The platform works by allowing users to create "channels" where each channel can have up to eight fields for different types of sensor data. Devices send data to these fields using standard HTTP requests or MQTT protocols. Once data is received, it is stored in real-time and can be accessed through graphs, charts, or even embedded dashboards. Users can perform real-time analysis or trigger actions using MATLAB code, which is natively integrated within the platform. ThingSpeak is widely used for environmental monitoring, smart agriculture, home automation, and industrial applications. The types of data typically displayed include temperature, humidity, air quality, pressure, soil moisture, GPS coordinates, energy consumption, water levels, and more. These data points are presented through customizable visualizations like line graphs, bar charts, or numeric displays, enabling users to monitor trends, detect anomalies, and make informed decisions based on live and historical data. The platform also supports triggering web-based alerts and sending data to other services, making it a powerful tool for both simple DIY projects and complex IoT deployments.

Energy-efficient Routing Protocols

Energy-efficient routing protocols in Mobile Ad Hoc Networks (MANETs) are developed to preserve the limited battery resources of mobile nodes while maintaining dependable data delivery in networks with constantly shifting topologies. These protocols strive to lower energy usage associated with data transmission, idle listening, and control message exchange. A common strategy is duty cycle sleep scheduling, in which nodes switch between active and low-power sleep states according to predefined schedules or triggered events, thereby reducing unnecessary power drain during idle periods. One notable example is the TEEN (Threshold- sensitive Energy Efficient sensor Network protocol), a reactive routing approach suitable for time-sensitive scenarios. TEEN operates by defining two thresholds—hard and soft—for sensor readings, ensuring nodes transmit data only when these thresholds are crossed, which minimizes transmissions and conserves energy. Overall, such protocols are essential for prolonging the network's operational duration, particularly where battery replacement or recharging is impractical. In IoT-integrated MANETs, energy- efficient routing is equally vital for maintaining reliable and sustainable communication among smart devices, especially in settings without fixed network infrastructure. Given that IoT nodes typically rely on finite power sources like batteries or harvested energy, optimizing energy use is of utmost importance. These protocols, integrated with duty cycle sleep scheduling, help manage the power usage of sensors and actuators by allowing them to sleep when not in use and wake only when necessary for data transmission or reception. Protocols like TEEN are especially suitable for event-driven IoT applications (e.g., environmental monitoring, disaster detection), where data is transmitted only when significant changes occur, thereby reducing unnecessary communication. By intelligently managing routing paths, node participation, and transmission frequency, energy-efficient routing ensures the longevity and responsiveness of IoT-enabled MANETs, making them scalable and robust for smart cities, remote health monitoring, and industrial automation scenarios.

Implementing Protocols using NS3

To implement energy-efficient routing protocols like TEEN with duty cycle sleep scheduling in NS-3 and visualize them in NetAnim and graphs, developers must first model the custom protocol behavior within the NS-3 simulation environment. This involves programming the protocol logic—such as threshold checks, energy-aware routing decisions, and node sleep-wake cycles—using C++ or Python APIs in NS-3. Energy models are attached to nodes to monitor power consumption dynamically. Once the simulation is configured, NetAnim can be used to visually represent node mobility, packet flow, and sleep states over time by generating an XML trace file from the NS-3 script. Additionally, energy usage statistics and performance metrics (e.g., packet delivery ratio, lifetime, delay) can be collected using NS-3's in-built tracing and logging tools, and then plotted as graphs using Gnuplot or Python libraries like Matplotlib. This combined visualization and analysis allow researchers to evaluate the effectiveness of energy-efficient protocols in IoT-enabled MANET scenarios.

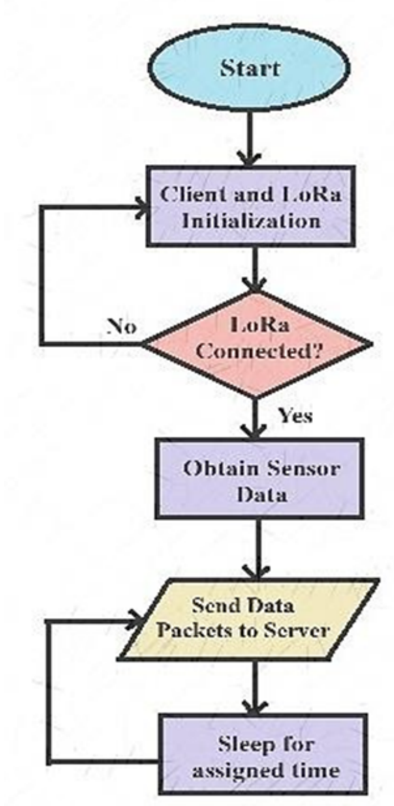

Fig 3. Flow Chart of Client Node

Diagram of the Client Node flow is shown in Figure 3. The client node starts by initializing its internal modules, including the microcontroller, connected LoRa module (RA-02), and the respective sensors (e.g., temperature, gas, etc.). This initialization ensures that the system is powered and ready to communicate. Once initialization is complete, the system checks for a successful connection to the LoRa module. If the LoRa is not connected, the system loops back to retry the initialization process. Once the LoRa connection is established, the client node proceeds to collect environmental or sensor data from the connected sensors. This data is packaged into a suitable format to be transmitted efficiently. The prepared data packets are then sent via the LoRa module to the server node. After successfully transmitting the data, the client node enters a low-power or sleep mode for a predefined time to save energy. This sleep-wake cycle is critical in ensuring energy efficiency and extended battery life in the client nodes deployed in remote or outdoor environments.

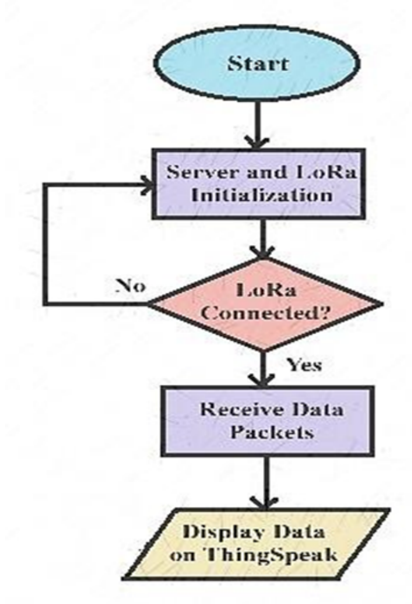

Fig 4. Flow Chart of Server Node

The server node begins its operation by initializing the LoRa module and microcontroller unit, just like the client. It prepares itself to receive incoming data packets from the client nodes. Once the setup is complete, the system checks whether the LoRa module is connected and operational. If the LoRa is not connected, the system reinitializes. Upon successful connection, the server continuously listens for incoming packets. When a data packet is received, the server node parses and decodes the sensor information. This data is then processed and sent to the cloud using a GSM module or Wi-Fi module, where it is displayed on a dashboard like ThingSpeak for real-time monitoring. Additionally, NS-3 simulations with NetAnim and Xgraph modules may run in parallel on the server side (offline or virtually) to evaluate energy efficiency and packet routing performance using similar parameters. These simulations help in analyzing and optimizing the protocol's performance even though the physical data is collected from actual hardware nodes.

System Design

Figure 5 shows the architectural diagram of the proposed Energy efficient Routing protocol using IoT-enabled Manets. The system design illustrated in the block diagram represents a LoRa-based forest fire detection and monitoring network comprising three client nodes (ESP32) deployed in a forest-fire- prone area. Each client node is equipped with a temperature and humidity sensor, smoke sensor, flame sensor, and GPS module to monitor environmental conditions. These client nodes are connected to individual LoRa modules to wirelessly transmit real-time sensor data. The collected data is sent to a centralized server node (ESP32) via LoRa communication. The server node is interfaced with a GSM module for remote communication and is connected to a software system including ThingSpeak (for IoT data visualization), NS3, and NetAnim (for network simulation and analysis). This integrated setup enables real- time environmental monitoring, early fire detection, and performance analysis of the communication protocol, ensuring both practical deployment and simulation-based validation.

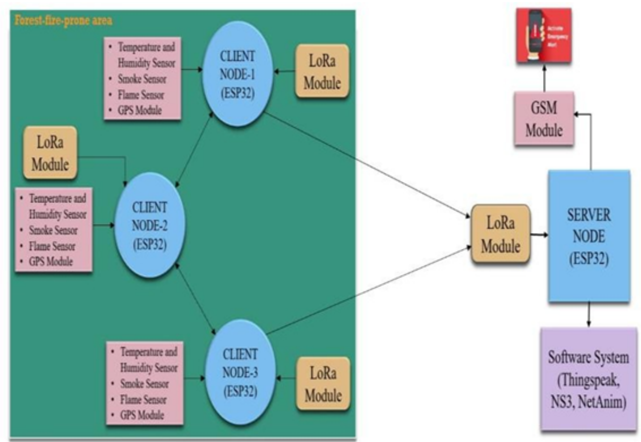

Fig 5. Architectural Diagram of the system

Fundamentally, the system is designed as a wireless sensor network using LoRa communication to detect forest fires in real time. Multiple ESP32-based client nodes are deployed across a forest area, each embedded with key sensors—temperature, humidity, smoke, flame, and GPS—to monitor environmental parameters indicative of fire. These nodes communicate wirelessly through LoRa modules, transmitting data to a central server node (also ESP32) that acts as a coordinator. The server processes and forwards the data using a GSM module for remote alerts and interfaces with a software system (ThingSpeak for visualization, NS3 and NetAnim for simulation and analysis). This distributed, low-power architecture ensures long-range communication, real-time monitoring, and simulation support for testing and optimization.

Result

The results of the system demonstrate effective real-time forest fire detection and monitoring with reliable wireless communication between client and server nodes using LoRa technology. Each client node successfully collected environmental data through its sensors detecting temperature spikes, smoke presence, and flame occurrence while accurately reporting

GPS locations. The LoRa modules enabled long-range, low-power data transmission without significant packet loss or delays, even across challenging terrain. The server node received the sensor data seamlessly, triggering alerts through the GSM module when fire conditions were detected. Integration with ThingSpeak allowed real-time visualization of sensor readings, and simulation tools like NS3 and NetAnim validated the communication network's performance under different conditions. Overall, the system proved to be robust, energy- efficient, and suitable for deployment in remote forest areas for early fire detection and alarm.

Fig 6. Result of ThingSpeak Sensor Values

In Figure 6, the result is shown in Server Node. The values of each sensor- DHT11, MQ9 and KY026 is displayed which are sent by the Client Node via LoRa module as data packets to Server Node. An alert icon is used to show the status of Fire Detection.

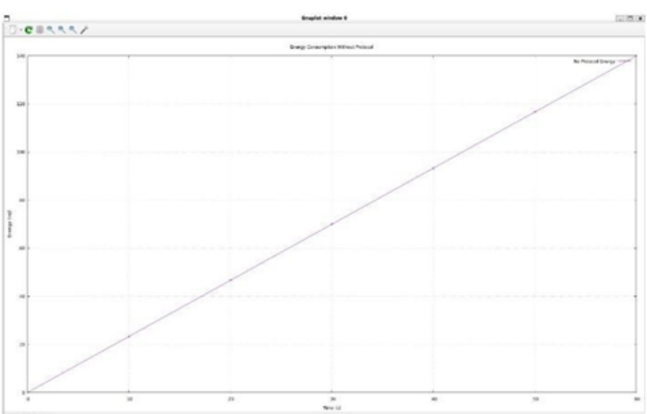

Fig 7. Graph before using protocols

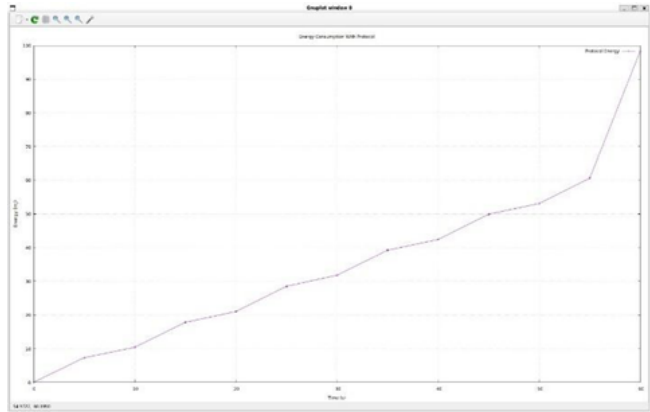

Fig 8. Graph after using protocols

Figure 7 shows the Energy versus Time Graph of Client Node in X-Graph without using Energy-efficient Protocols. While Figure 8 shows the same but using the Energy-efficient Protocols like TEEN and Duty Sleep Cycling Protocols.

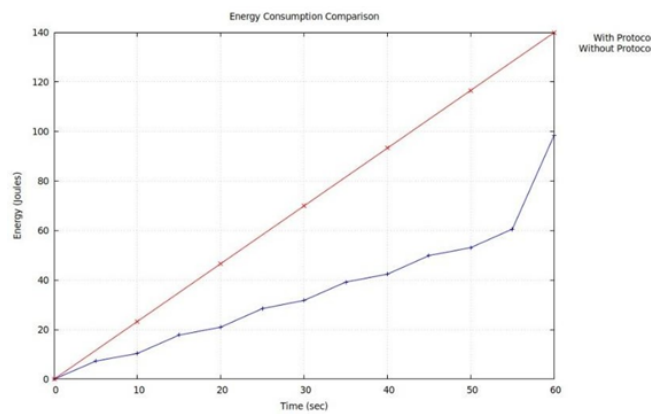

Fig 9. Comparison of both Graphs

Here, in Figure 9 We have the comparison of both the graphs. The red one representing energy consumed while using protocols, and the blue line representing energy consumed without using protocols. We get the concise idea of how these protocols help in reducing the energy consumption of the Client Node.

The Figure 10 is represented using NetAnim software, basically used to show the node-to-node communication and the data packet flow between them. We considered three ESP32 Client Nodes, that act as a MANET and are placed at fixed distances, which we assume will be inside the forest fire-prone area. The Server node is capable of receiving data packets at regular intervals from the Client Nodes through wireless communication using LoRa module that communicate with each other.

Fig 10. Data Packet flow between Nodes

Simulation Metrics

In this project, Energy vs Time was chosen as the primary simulation metric to evaluate the effectiveness of the energy-efficient routing protocol used in the Wireless Sensor Network (WSN). This metric reflects how much power is consumed over a specific period while the sensor nodes perform data collection, transmission, and idle/wake operations. The simulation was conducted using NS-3, with results visualized through Xgraph and NetAnim tools to track energy levels across nodes dynamically. The energy usage was recorded per second, which allowed us to measure the impact of AODV routing combined with MAC duty cycling protocols in reducing overall energy drain. This is crucial in remote forest environments where power sources are limited and battery-operated nodes must operate autonomously for extended durations.

Now we calculate real-time energy consumption per module based on our project setup where a Li-Ion battery powers the client node, and we are not using energy-efficient protocols (so modules stay in active mode). We'll assume a standard Li-Ion battery voltage of 3.7V (nominal), and a 1-minute operation period (60 seconds) as the basis for energy calculation.

Energy Consumption Formula:

Energy (E) = V x I x t

Where:

E = Energy in joules (J)

V = Voltage in volts

I = Current in amperes

t = Time in seconds

Table 2 shows the calculated values of each module on the Client Node that include their voltage, current and an assigned time for each of them. Thus, the total Energy consumed by the Client Node without using any protocols is 139.914 Joules per minute.

Table 1. Normal Voltage and Current Consumptions of Modules

Module	Operating Voltage (V)	Current Consumption
ESP32	3.0 – 3.6 V (typ. 3.3 V)	160–240 mA, 20 mA (Idle), <1 mA (Deep Sleep)
LoRa RA-02 (SX1278)	1.8 – 3.7 V (typ. 3.3 V)	~120 mA, <1.5 mA (Sleep)
DHT11 Sensor	3.3 – 5.5 V	~0.5 – 2.5 mA (active read)
MQ-9 Gas Sensor	5.0 V	~150 – 200 mA
KY-026 Flame Sensor	3.3 – 5 V	~20 – 30 mA
Neo-6M GPS Module	3.0 – 5.0 V (uses 3.3V internally)	~30 – 45 mA (active), <10 mA (standby)
Li-Ion Battery	3.7 V nominal (4.2V full)	Supplies current based on load

Table 2. Calculated values before Protocol implementation

Module	Voltage (V)	Current (A)	Time (s)	Energy (J)
ESP32 (WiFi+Active)	3.7	0.2	60	44.4
LoRa RA-02 (Tx)	3.7	0.12	60	26.64
DHT11	3.7	0.002	60	0.444
MQ-9 Gas Sensor	5	0.18	60	54
KY-026 Flame	3.7	0.025	60	5.55
Neo-6M GPS	3.7	0.04	60	8.88
Total				139.914

Table 3 shows the calculated values of each module in Client Node and their Power states after applying the energy-efficient protocols, such as TEEN and Duty Sleep Cycling Protocols.

Hence, here we compute all these values of each module together to get the overall value for the Client Node.

Total ≈ 139.91 J over 60s

Table 4 shows the values of overall calculated energy of the Client Node with respect to Time. We can see the increase in the Energy consumption of the Node as the time increases. At the end, full energy was consumed by the node.

Table 5 shows the values of overall calculated energy of the Client Node with respect to Time. We can see the varying increase in the Energy consumption of the Node as the time increases, but it is varying because of the energy-efficient protocols that we have used. At the end, the full energy usage was conserved and thus it decreased to 98.4 Joules, reducing Energy consumption of the overall Node.

Table 5. Overall

Time (s)	Energy Used (J)	Remarks
0	0	Start
5	7.3	LoRa + ESP32 wake (burst), sensors active
10	10.4	Light draw (sensors only)
15	17.8	Burst again (LoRa + ESP32), GPS active
20	21	Mostly sensors
25	28.5	Burst (ESP32+LoRa wake)
30	31.8	Flat
35	39.2	Burst again
40	42.4	Light power draw
45	49.9	ESP32 + LoRa wake again, GPS active
50	53.1	Sensors only
55	60.6	Final burst before end
60	98.4	Total accumulated (ESP32 + LoRa only active ~5.5s total)

Table 3. Calculated

Module	Operating Voltage (V)	Current Consumption	Power State
ESP32	3.3	160-240 mA , ~20 mA e), <1 mA (Deep Sleep)	Active / Idle / Sleep
LoRa RA-02	3.3	~120 mA (Tx), ~12 (Rx), <1.5 mA (Sleep)	Transmit / Receive / Sleep
DHT11	5	~0.5-2.5 mA (active)	Idle most of the time
MQ-9 Gas Sensor	5	~150-200 mA	Continuous draw
KY-026 Flam Sensor	5	~20-30 mA	While detecting
Neo-6M GPS Module	3.3	~30-45 mA (active), <10 mA (standby)	Cold start and ctive acquisition
Li-Ion Battery (18650)	3.7	Depends on load	Power supply

Table 4. Overall

Time (s)	Energy Used (J)	Remarks
0	0	Start
10	23.32	Continuous use
20	46.64	Linear growth
30	69.96	
40	93.28	
50	116.59	
60	139.91	Full energy consumed

Recommendations for Further Research

To further advance the system, important research avenues involve incorporating machine learning techniques to boost the accuracy of fire predictions and minimize false alarms, enhancing energy efficiency through solar energy or intelligent power management methods, and expanding the communication range by utilizing LoRa mesh networking. Additionally, incorporating real-time image or video surveillance can enhance situational awareness, make the solution more scalable and practical for real-world deployment.

Conclusion

In this work, we develop a forest fire detection and alert system that utilizes a network of ESP32 nodes integrated with multiple sensors, providing a reliable and effective solution for early fire monitoring. By integrating temperature, smoke, and flame sensors with GPS and GSM modules, the system ensures timely alerts to nearby firemen, facilitating rapid response to potential fire incidents. The use of energy-efficient protocols and components, such as the LoRa module for extended communication range, enhances the system's reliability and operational longevity in remote areas. Overall, this system represents a significant step towards improving forest management and safety, helping to mitigate the risks associated with forest fires.

References

1) Khan MA, Khan A. Energy-efficient routing protocols for IoT-enabled mobile ad hoc networks: A review. *Journal of Network and Computer Applications*. 2021;175:102905. Available from: https://doi.org/10.1016/j.jnca.2020.102905.
2) Sharma S, Jain A. A survey on energy- efficient routing protocols for mobile ad hoc networks. *Wireless Networks*. 2020;26(3):1239–1254. Available from: https://doi.org/10.1007/s11276-019-02130-0.
3) Javed MY, Raza MA. A novel energy- aware routing protocol for IoT-based MANETs. *Future Generation Computer Systems*. 2021;113:515–528. Available from: https://doi.org/10.1016/j.future.2020.08.020.
4) Bashir A, Khan I. A comprehensive review of energy efficient routing protocols in mobile ad hoc networks. *Ad Hoc Networks*. 2020;105:102183. Available from: https://doi.org/10.1016/j.adhoc.2020.102183.
5) Bhardwaj A, Kaur M. Survey of energy- efficient routing protocols in mobile ad hoc networks. *Computers & Electrical Engineering*. 2021;92:107145. Available from: https://doi.org/10.1016/j.compeleceng.2021.107145.
6) Patel H, Kotecha K. Energy-Efficient Routing in IoT and MANETs: A Comprehensive Survey. *IEEE Access*. 2020;8:22006–22030. Available from: https://doi.org/10.1109/ACCESS.2020.2969937.