PROJECT PROPOSAL

SmartGuard : Simulated IoT Environmental Monitoring

Team Members
Chepuri Venkata Naga Thrisha (23110079) Revathi Katta (23110159)
Sai Keerthana Pappala (23110229) Thoutam Dhruthika (23110340)

1. Problem Statement

In critical spaces like classrooms and labs, the physical environment directly impacts the health, safety, and productivity of its occupants. The old way of monitoring, like having someone manually check rooms or using offline devices, has two big problems. First, it **requires a lot of manual work** and **doesn't give you live information**. This means a serious issue, like a buildup of CO_2, could be missed for hours. Second, it's **hard to manage this for many rooms at once**. The data collected is often disconnected and stored in different places, making it impossible to see overall patterns or trends.This project addresses the challenge of designing a centralized IoT monitoring system, focusing specifically on creating a scalable and efficient network architecture that can reliably handle data from hundreds of sensors.

The core of this project is to simulate this network using **Mininet** to validate design choices before any hardware deployment. We will implement and analyze the **MQTT protocol**, a lightweight standard for IoT, and measure key performance metrics like latency and throughput to create a robust, scalable blueprint for a real-time monitoring system.

2. Objectives

The primary goals of this project are:

a) To design and simulate a scalable network topology in Mininet for at least 100 sensor clients.
b) To implement a reliable data pipeline using the MQTT protocol.
c) To develop a backend service to process and store time-series data in InfluxDB.
d) To create a centralized Grafana dashboard for real-time visualization and historical analysis.
e) To analyze the network's performance by measuring latency, throughput, and packet loss.

3. System Architecture and Methodology

Our methodology is structured around a clear, four-stage pipeline, progressing from network simulation to data analysis. This approach ensures each component is well defined and can be tested systematically.

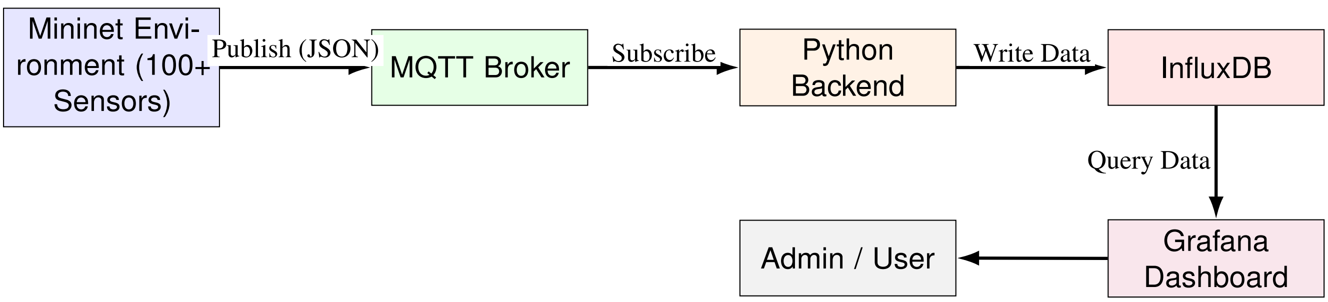

Figure 1: High-Level System Architecture of the SmartGuard Monitoring System.

3.1 Network Simulation and Data Generation

The project's foundation is a virtual network created in **Mininet**. We will design a custom tree topology with a central router, multiple switches, and over 100 host nodes to realistically model a large-scale deployment. Each host will run a Python script acting as a simulated sensor, generating periodic environmental data (e.g., temperature, humidity) to create a consistent data stream for testing the system under load.

3.2 Communication Protocol and Data Flow

We selected **MQTT** for its lightweight, publish-subscribe model, which is ideal for IoT networks. It allows multiple clients to efficiently transmit small, frequent messages to a central server with low overhead and reliable delivery. While other protocols exist, they are less suitable for this project: **HTTP** is too heavy for continuous streaming; **WebSockets** and **CoAP** add unnecessary complexity or have limited tooling; and enterprise systems like **AMQP** or **Kafka** are overly complex for this scale. Therefore, **MQTT** provides the optimal balance of efficiency, scalability, and simplicity for our monitoring system.

3.3 Backend Processing and Time-Series Storage

A central backend service, developed in **Python**, will subscribe to the MQTT broker to ingest all sensor data in real-time. Upon receiving a message, the service will parse the JSON payload, add a server-side timestamp for accuracy, and write the structured result to an **InfluxDB** database. We chose InfluxDB specifically because it is a high-performance time-series database, optimized for the fast ingestion and complex querying required for monitoring applications.

3.4 Visualization, Alerting, and Performance Evaluation

The final stage makes the collected data actionable. A **Grafana** dashboard will serve as the centralized user interface, connected directly to InfluxDB. It will provide visualizations such as time-series graphs to track historical trends, gauges for at a glance readings, and heatmaps to compare conditions across all rooms. An alerting system will be configured to notify users of anomalies, like high CO_2 levels.

Crucially, we will also perform a rigorous network evaluation. Using tools like Wireshark and iPerf, we will measure the three core performance metrics **latency**, **throughput**, and **packet loss** while progressively increasing the number of active sensors to validate the scalability and robustness of our architecture.

4. Tools and Technologies

- **Network Simulation:** Mininet
- **Programming Language:** Python
- **Communication Protocol:** MQTT (using the Mosquitto Broker)
- **Time-Series Database:** InfluxDB
- **Dashboard & Visualization:** Grafana
- **Network Analysis:** Wireshark, iPerf

5. Expected Outcomes and Applications

Upon completion, this project will deliver:

- A fully functional simulation of an environmental monitoring IoT network capable of supporting over 100 clients.
- A real-time dashboard that visualizes environmental parameters and alerts staff to anomalous conditions.
- A comprehensive performance analysis report detailing the network's latency, throughput, and reliability under various loads.

The applications of this system are practical and impactful. It can be used to enhance the learning environment, improve energy efficiency by optimizing HVAC systems, and provide proactive safety monitoring in sensitive labs. The scalable architecture serves as a blueprint for a future real-world hardware implementation on campus.

References

[1] Lantz, B., Heller, B., & McKeown, N. (2010). A network in a laptop: Rapid prototyping for software-defined networks. *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks.*

[2] Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., & Ayyash, M. (2015). Internet of Things: A survey on enabling technologies, protocols, and applications. *IEEE Communications Surveys & Tutorials, 17*(4), 2347-2376.

[3] Hillar, G. C. (2015). *MQTT Essentials - A Lightweight IoT Protocol.* Packt Publishing.