Team 26 Homework 3 Report

Student IDs: s307798, s308347, s319928

Abstract

This report delves into the use of MQTT and REST protocols in IoT systems. The focus is on their application for data collection and management, highlighting the benefits and challenges associated with each approach and their suitability for IoT systems' dynamic requirements.

Exercise 1: Data Collection, Communication, and Storage

MQTT vs. REST

REST protocols, while common for web services, are not optimized for the sporadic connectivity and low bandwidth typical of IoT environments. In contrast, MQTT is specifically crafted for IoT applications, offering a lightweight protocol that conserves bandwidth and supports intermittent network connections.

MQTT Overview

MQTT stands out in IoT for its efficient publish-subscribe model, which facilitates real-time communication without the need for continuous connections. This model greatly reduces network overhead and enhances message delivery efficiency, making it ideal for the responsive nature of IoT ecosystems.

Summary

MQTT is preferable for IoT applications, surpassing REST with its efficient use of bandwidth, reliable performance in variable connectivity, and energy-saving capabilities. Its design aligns with the operational demands of IoT devices, establishing it as the superior choice for such applications.

Exercise 2: Data Management and Visualization

RESTful API Design

Our 'cherrypy' RESTful API adeptly handles IoT data with endpoints for device inventory and data management.

Method	Endpoint	Desc.
GET	/devices	List devices.
GET	/device/{mac}	Get device data.
DELETE	/device/{mac}	Del device data.

Table 1: API Endpoints

HTTP Method Overview

1. GET Requests data without affecting the resource.

2. POST Not used in this API.

3. PUT Not applicable to this API.

4. DELETE Removes device data, aligning with privacy standards.

Method Descriptions

1. GET /devices Lists all devices for inventory checks.

2. GET /device/{mac} Retrieves data for device monitoring.

3. DELETE /device/{mac} Securely erases a device's data for data integrity.