Class: ITAI 3377

Name: Win Aung

Team: AI Master

Date: 6.8.2024

Professor: Mr. Tawanda

**Reflective Journal: IIoT Protocols Project**

**Introduction**

This project focused on exploring key communication protocols used in the Industrial Internet of Things (IIoT), including **MQTT**, **CoAP**, and **OPC UA**. The goal was to understand how these protocols operate in industrial environments, how they differ from each other, and how they are implemented in real-world systems.

At the start of the project, my personal goal was to learn how lightweight communication works between devices and to gain hands-on experience with protocols that support real-time and efficient data exchange. I was also curious to see how edge devices like sensors or controllers could use these protocols for reliable communication.

**Personal Contributions**

I was responsible for setting up and testing the **MQTT protocol** using a simulated smart environment in Visual Studio Code. I also helped configure client-broker communication and validate message delivery using a lightweight Python-based MQTT library (paho-mqtt). In addition, I contributed to the documentation of the **CoAP section**, including how it compares to MQTT in terms of simplicity and reliability.

In terms of code, I wrote the full MQTT script to demonstrate device-to-device messaging, and I supported the integration of CoAP-based communication in another group member's test setup. I also provided visual diagrams that were included in our final report to explain how these protocols work in IIoT systems.

**Learning Outcomes**

**🔹 MQTT (Message Queuing Telemetry Transport):**

I learned that MQTT is a publish-subscribe-based messaging protocol ideal for low-bandwidth, high-latency networks. It is widely used in IIoT because of its simplicity, reliability, and support for persistent sessions.

**🔹 CoAP (Constrained Application Protocol):**

CoAP is a request/response protocol designed for use with constrained devices and networks. I discovered that it works similarly to HTTP but is much lighter, making it suitable for small sensors and actuators. It was interesting to see how it uses UDP instead of TCP.

**🔹 OPC UA (Open Platform Communications Unified Architecture):**

OPC UA was the most complex protocol in the project. I learned that it's widely used in industrial automation because of its strong data modeling capabilities and built-in security. Unlike MQTT and CoAP, it supports structured data, access control, and real-time monitoring.

**Challenges and Solutions**

One of the main challenges was setting up **communication between different protocols** in the same simulation. CoAP was especially tricky due to limited resources and debugging tools. I initially struggled with installing the proper libraries and setting up endpoints.

To resolve this, I searched for official documentation, consulted classmates, and used online forums. I also tested the CoAP server using Postman and a Python-based client to verify responses. Another challenge was understanding OPC UA’s complex structure and configuration. We overcame this by breaking it into simpler components and testing them one at a time.

**Future Applications**

The knowledge I gained from this project will be highly useful in future IIoT, smart building, or industrial automation projects. I now understand which protocol to use depending on the device constraints and communication needs.

For example, I would choose MQTT for real-time monitoring in smart agriculture, CoAP for energy-efficient sensor networks, and OPC UA for complex, secure factory systems. As a next step, I would like to expand the project by combining multiple protocols in a single edge simulation to study interoperability and system scalability.