**Report: MQTT, CoAP, and OPC UA in an IIoT Sensor Network**

**Name:** Eduardo Cabrera-Lopez  
**Course:** ITAI 3377  
**Assignment:** Lab 04 — Conceptual Design of an IIoT Sensor Network  
**Date:** June 17, 2025

**Overview**

Using three industrial communication protocols—MQTT, CoAP, and OPC UA—in this lab I created and simulated an IIoT sensor network. To grasp their behavior, efficiency, and implementation variations, real-time temperature and humidity data was generated and sent utilizing every protocol.

**🔌 Protocol Comparison**

| **Feature** | **MQTT** | **CoAP** | **OPC UA** |
| --- | --- | --- | --- |
| **Type** | Publish/Subscribe (asynchronous) | Request/Response (HTTP-like) | Client/Server (object-based) |
| **Transport** | TCP | UDP | TCP (binary) |
| **Message Format** | JSON | JSON (via payload) | Binary object model |
| **Ease of Setup** | Very easy with Mosquitto broker | Moderate (aiocoap has issues on Windows) | Moderate to advanced (requires asyncua) |
| **Performance** | Fast, low-overhead, great for IoT | Lightweight, good for constrained devices | Rich data model, best for industrial systems |
| **Use Case** | Ideal for fast telemetry and alerts | Ideal for sensor queries or low-power IoT | Best for structured industrial automation |
| **Visualization** | Easy to integrate and plot live | Troublesome on Windows (UDP fallback used) | Runs well, but harder to visualize |

**Simulation**

**MQTT**

Using Matplotlib, data was received and visualised in real time; paho-mqtt was used to publish temperature and humidity readings every second.   
•. There were no main problems; the outcomes were consistent and repeatable.

**CoAP**

• tried using aiocoap but ran against ongoing platform-specific Windows (e.g., [WinError 10049]) problems.   
• I replicated CoAP-style behavior using raw sockets in a fallback UDP simulation instead.   
•. Not perfect but still satisfied the simulation target.

**OPC UA**

On a simulated OPC UA server, asynchronous to expose variables; temperature and humidity values were updated live and could be seen using tools like UaExpert or logged in the terminal.   
• Though it was more difficult, configuration worked great once some asynchronous handling problems were fixed.

**Visualization Summary**

• The only protocol clearly shown with a dynamic Matplotlib display was MQTT.

* Timestamps were changed and structured to resolve axis scaling problems.
* To record this procedure, screenshots were taken (mqtt\_visualisation.png).

**Key Learnings**

* Ideal for low-power or real-time IoT data streaming, MQTT is quick, simple to use.
* On paper, CoAP is fantastic; on some platforms—especially Windows—it can be erratic.
* Though it has a longer learning curve, OPC UA is incredibly strong and scalable for industrial use.

**Conclusion**

This session exposed me practically to three fundamental communication technologies in industrial IoT. Every one of them has advantages based on the system requirements. MQTT was the most easily available for simple setup and visualizing. OPC UA is the greatest long-term option for scalable object-oriented industrial systems. In lightweight sensor networks, CoAP is still useful, but deployment should take implementation hurdles into account.