CPE 314: Computer Networks (2/65)
Project I

MQTT (MQ Telemetry Transport) is a popular application layer protocol for Internet of Things (IoT) applications. The communication model is the published-subscribed pattern, which consists of three node roles – Publisher, Subscriber, and Broker. A publisher publishes its topic to Broker and one or more subscribers subscribe for the topic. Whenever the publisher publishes data, Broker will automatically relay it to all nodes subscribed to that topic.

A group of *four to six students* is to design and implement a MQTT-based IoT application that sends sensor readings from an IoT node to a remote database. The system consists of three entities – Client (IoT node), Broker, and Server. The system architecture is shown in Fig.1, where each entity works as following:

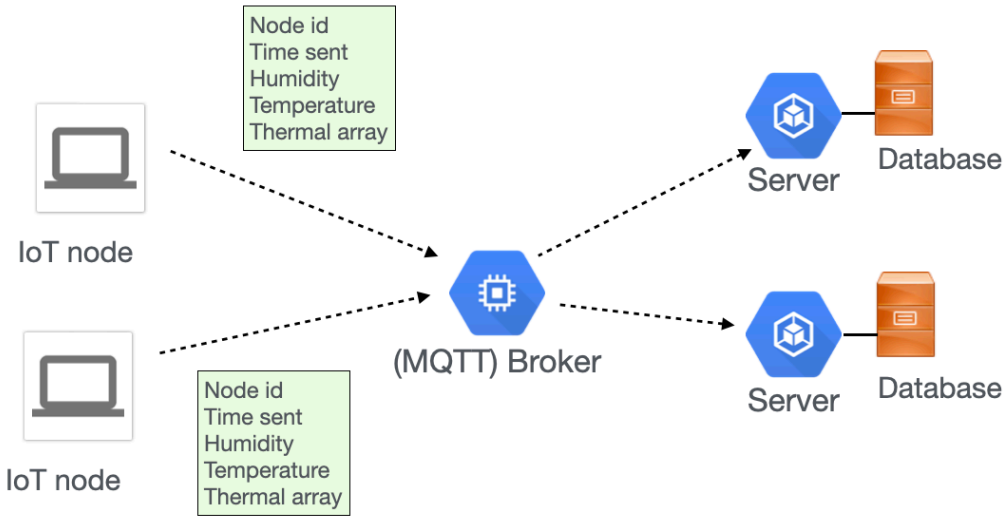

Figure 1: System architecture

- Client has three types of sensors – Relative humidity, temperature, and thermal array. The relative humidity readings are between 0 to 100 percent, the temperature readings are between 0 to 90 degree celsius, and the thermal array readings are 24×32 where each array value represents a temperature reading between 5 to 60 degree celsius. Client simultaneously reads all the sensors every 3 minutes and wants to send the sensor data to Broker together with 4-digit node id and current time (Date, hours, minutes). To emulate the sensor readings without installing real sensors, Client can read sensor data stored in an Excel file and sends to Broker. The sample input is shown in Fig.2.
- Broker forwards any data it receives to its subscribers.
- Server subscribes for data from Client. The received data is assembled as necessary and written in the local database (e.g., MySQL, PostgreSQL) for later query or visualization. Data sent in the same round from Client must be stored in the same database record. Any lost data would result in missing values in the database records.

The application works under the following constraints and requirements:

	A	B	C	D
1	Time	Humidity	Temperature	ThermalArray
2	2022-11-08 12:08:00	55.82	29.16	31.5,31.6,31.9,32.0,31.6,30.9,31.5,31.4,31.1,31.1,31.2,31.2,31.3,31.2,31.0,31.1,31.5,30.7,
3	2022-11-08 12:10:00	55.25	30.7	32.4,32.2,32.6,32.2,32.5,32.0,31.6,31.9,32.0,31.4,31.4,31.9,31.9,31.7,31.9,31.6,31.9,31.2,
4	2022-11-08 12:12:00	55.33	31.7	32.7,32.4,32.7,32.4,32.9,31.8,32.4,32.2,32.8,31.9,32.5,32.0,31.9,31.8,32.3,31.9,32.0,31.5,
5	2022-11-08 12:14:00	54.67	32.52	33.0,33.1,32.8,32.9,33.0,32.5,32.5,32.7,32.5,32.4,32.3,32.3,32.1,32.5,32.3,32.6,32.1,32.0,
6	2022-11-08 12:16:00	52.48	33.06	33.3,33.1,33.8,33.1,33.3,32.8,32.9,33.1,33.0,32.7,32.8,32.6,33.0,32.5,32.7,32.9,32.7,32.1,
7	2022-11-08 12:18:00	50.65	33.66	34.0,33.5,33.6,33.6,33.5,33.0,33.5,33.2,32.7,32.5,32.9,32.9,33.1,32.7,33.0,33.0,33.0,32.6,
8	2022-11-08 12:20:00	49.02	34.05	34.2,33.2,33.7,33.3,33.5,33.2,33.1,33.6,32.9,32.8,33.0,33.1,32.9,32.8,33.4,32.9,32.8,32.4,
9	2022-11-08 12:22:00	48.91	34.44	33.7,33.7,33.5,33.8,33.6,33.7,33.4,33.3,33.0,33.0,33.2,33.7,33.0,33.4,33.3,33.5,33.1,32.9,
10	2022-11-08 12:24:00	48.22	34.74	33.8,34.1,34.1,34.4,34.4,33.6,33.9,33.6,33.4,33.4,33.4,33.5,33.7,33.4,34.0,33.1,33.1,33.2,
11	2022-11-08 12:26:00	48.03	35.04	34.6,34.3,34.4,34.6,34.0,33.6,34.0,34.0,33.6,33.4,33.6,33.8,33.6,33.4,33.9,34.1,33.3,33.1,
12	2022-11-08 12:28:00	47.9	35.34	34.4,34.5,34.1,34.6,34.5,34.0,34.3,34.0,33.7,33.8,33.8,33.8,33.6,33.4,33.7,33.7,33.8,33.3,
13	2022-11-08 12:30:00	47.71	35.61	34.7,34.2,34.8,34.4,34.4,34.2,34.2,34.1,34.0,33.8,34.1,33.7,33.8,33.9,34.3,34.2,33.9,33.2,
14	2022-11-08 12:32:00	47.22	35.89	34.8,34.6,34.7,34.5,35.2,34.2,34.2,34.2,34.2,33.8,34.0,33.8,34.2,34.0,34.1,33.6,34.1,33.7,
15	2022-11-08 12:34:00	47.21	36.15	34.7,34.7,34.6,34.8,34.4,34.3,34.5,34.5,34.3,34.1,33.8,34.1,34.1,34.1,34.2,34.4,34.0,33.8,
16	2022-11-08 12:36:00	47.02	36.25	35.1,35.1,34.9,35.0,35.0,34.6,34.4,34.7,34.6,34.4,34.1,34.4,34.2,34.4,34.4,34.4,34.2,34.3,
17	2022-11-08 12:38:00	46.83	36.34	35.5,35.0,34.9,35.1,34.7,34.8,34.6,34.8,34.5,34.2,34.2,34.0,34.6,34.1,34.7,34.4,34.1,34.1,

Figure 2: Sample input sensor readings

- ▷ Client can only send at most 250 bytes in one message.
- ▷ Multiple IoT nodes and servers can be deployed in the system. If Server subscribes data from several clients, all the data is stored in the same database table.
- ▷ Broker prints out an IP address on the screen when a new subscriber or publisher connects or disconnects.
- ▷ Broker prints out published messages on the screen.
- ▷ Server prints out received messages from Broker on the screen.
- ▷ At Server, one must be able to query data of each sensor separately from the database.

The software can be coded in any programming language you like (Python, C, Java, etc.). You are allowed to use MQTT library APIs provided by the language.

Deliverables

By midnight of **Sunday March 5, 2023**, submit in one `.zip` file containing the final report, the source code (and executable files if any) and database setup configuration scripts, and the `readme` file describing what each file is. Put sufficient comments in the source codes so that the codes are readily comprehensible.

The final report must contain the program flowcharts and detailed explanation of operations executed by the three entities (Client, Broker, Server).

The grading is based on Program completeness and comprehensibility (60%), Report quality and writing (20%), and Presentation demo + Q&A (20%).

Demo

Sign up for a 10-minute demo in allocated time slots on March 9-10 (Sign-up sheet to be provided). Only one person in your group will be randomly chosen to demo, explain the codes, and answer questions.

All parts of the project must be your own group's work. Any form of copying from another group/source is an academic fraud and may receive zero credits as the highest penalty.