

IOT PROTOCOLS

IOT PROTOCOLS

1.MQTT

MQTT stands, for Message Queuing Telemetry Transport. It is a lightweight messaging protocol for use in cases where clients need a small code footprint and are connected to unreliable networks or networks with limited bandwidth resources.

- A MQTT is primarily used for machine-to-machine (M2M) communication or Internet of Things types of connections.
- MQTT implements the publish/subscribe model by defining the clients and brokers as below:

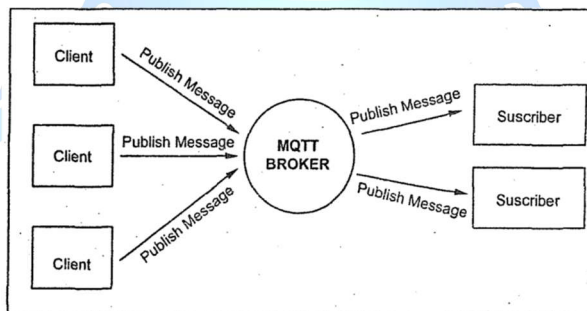


Fig 11.19 MQTT architecture

(i) MQTT client

- An MQTT client is any device from a server to a microcontroller that runs on an MQTT library. If the client is sending messages, it acts as a publisher, and if it is receiving messages, it acts as a receiver.
- Basically, any device that communicates using MQTT over a network can be called an MQTT client device.

(ii) MQTT broker

- The MQTT broker is the backend system which coordinates messages between the different clients. Responsibilities of the broker include receiving and filtering messages, identifying clients subscribed to each message, and sending them the messages. It is also responsible for other tasks such as:
 - ❖ Authorizing and authenticating MQTT clients.
 - ❖ Passing messages to other systems for further analysis.

(iii) MQTT connection

- Clients and brokers begin communicating by using an MQTT connection. Clients initiate the connection by sending a CONNECT message to the MQTT broker. The broker confirms that a connection has been established by responding with a CONNACK message. Clients never connect with each other, they connect only with the broker.

Advantages

The advantages of MQTT are,

(i) Lightweight and efficient

MQTT implementation on the IoT device requires minimal resources, so that it can even be used on small microcontrollers. MQTT message headers are also small so that network bandwidth can be optimized.

(ii) Scalable

MQTT implementation requires a minimal amount of code that consumes very little power in operations.

(iii) Secure

ROHINI COLLEGE OF ENGINEERING AND TECHNOLOGY
MQTT makes it easy for developers to encrypt messages and authenticate devices and users using modern authentication, protocols.

(iv) Well-supported

Several languages like Python have extensive support for MQTT protocol implementation. Hence, developers can quickly implement it with a minimal coding on any type of application.

2.XMPP

XMPP is a short form for Extensible Messaging Presence Protocol. It's protocol for streaming XML elements over a network in order to exchange messages and presence information in close to real time. This protocol is mostly used by instant messaging applications like WhatsApp.

Requirements: Functions

The following are the basic requirements of any Instant Messenger which are fulfilled by XMPP:

- (i) Send and receive messages with other users.
- (ii) Check and share presence status.
- (iii) Manage subscriptions to and from other users.
- (iv) Manage contact list

(v) Block communications (receive message, sharing presence status, etc) to specific users.

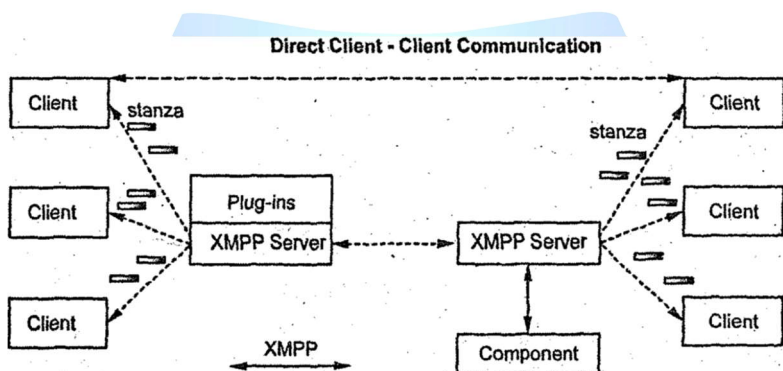
- **Decentralized:**

XMPP is based on client-server architecture, i.e. clients don't communicate directly, they do it with the help of server as intermediary. It is decentralized means there is no centralized XMPP server just like an email, anyone can run their own XMPP server.

- **Addresses In XMPP**

Addresses in XMPP are similar to standard e-mail addresses with a couple of notable differences. JIDs include an optional node, a domain, and an optional resource in the form:

[node " @ "] domain ["/" resource]



- In XMPP architecture, a client communicates with another client when both are in same domain. The servers can also communicate for the purpose of routing between domains.
- Each client implements the client form of the protocol, where the server provides routing capability. Gateways can exist for purposes of translating between foreign messaging domains and protocols.

3.MODBUS

- Modbus or MODBUS is a client/server data communications protocol in the application layer of the OSI model. It was originally published by Modicon in 1979 for use with its Programmable Logic Controllers (PLCs).
- Modbus is a request-response protocol implemented using a master-slave relationship. In the communication always occurs in pairs, one device must initiate a request and then wait for a response. The initiating device (the master) is responsible for initiating every interaction.

is responsible for initiating every interaction.

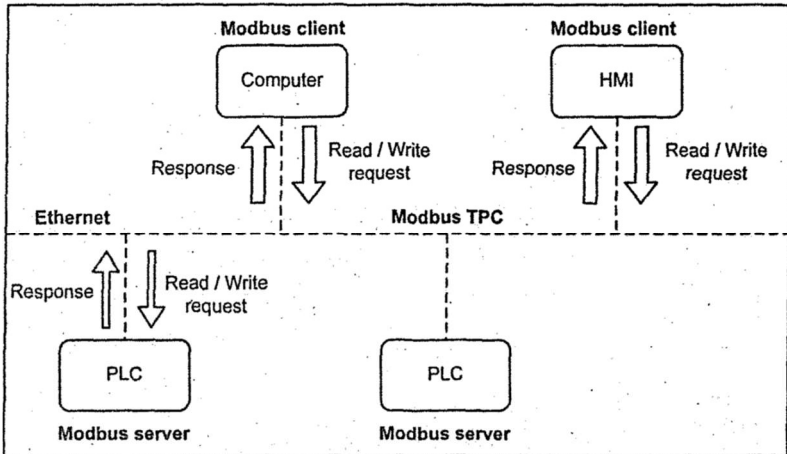
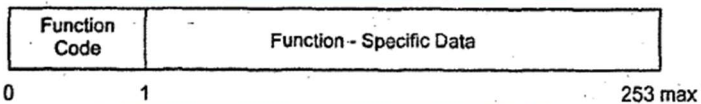


Fig 11.21 Modbus protocol

- The PDU consists of a one-byte function code followed by up to 252 bytes of function-specific data. The function code is the first item to be validated. If the



- The packet size is limited to 253 bytes, the most common function codes can transfer between 240 and 250 bytes of actual data from the slave data model, depending on the code.

- Modbus uses the Transmission Control Protocol (TCP), which provides connection-oriented communication, error detection, and flow control.

Advantages

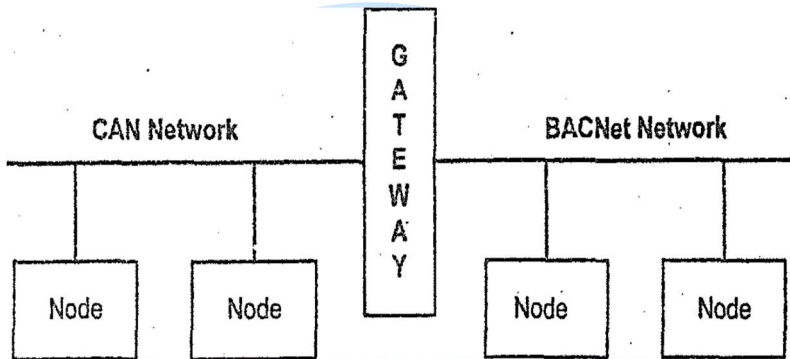
The advantages of Modbus are,

- (i) Simple but efficient technical architecture.
- (ii) Very easy integration with installations.
- (iii) High reliability.

4.CANBUS with BACNet

- The Controller Area Network (CAN bus) is a message-based protocol designed to allow the Electronic Control Units (ECUs) found in today's automobiles, as well as other devices, to communicate with each other in a reliable, priority-driven fashion.
- BACnet is a network protocol used in Building Automation Systems (BAS) to control the data exchange between different devices and components. BACnet stands for Building Automation and Control Network.
- The CAN to BACNet gateway(PG-103-194) give a solution as a powerful and outstanding processor that is

ROHINI COLLEGE OF ENGINEERING AND TECHNOLOGY completely flexible and productive as a building & industrial automation gateway, allowing integrators to easily connect devices to networks in a commercial buildings and industrial facilities.



- The gateway reads the data from the CAN channel and converts it to BACnet IP format. The messages can be filtered by their CAN message IDs, and the incoming bytes of data can be received.
- The BACnet IP can be set up as a client and data from the CAN interface can be written to the BACnet server.
- BACnet client devices can read data from the gateway which we received from CAN end if it's configured as a BACnet IP server.

Features:

- (i) Upto 1000 points can be interconnected.
- (ii) Integrated Serial Port (one wire)
- (iii) Faster data transmission while decreasing the network traffic.

