IoT Module 2

CO2	Explain the protocols used in IoT infrastructure.		
M2.01	Outline the protocols for IoT – Messaging, Transport, addressing and Identification	2	Understanding
M2.02	Interpret the messaging protocol MQTT and CoAP in IoT applications	4	Understanding
M2.03	Explain the transport protocols Li-Fi and BLE involved in IoT	4	Understanding
M2.04	Explain IPv4 and IPv6 Addressing	3	Understanding

Contents: Different protocols in data link, network, transport and application layers (overview only) - Messaging Protocol - MQTT, CoAP - Transport Protocol - BLE, Li-Fi - Addressing - IPv4, IPv6 - URI

(Ref: Vijay Madisetti, Arshdeep Bahga, Internet of Things: A Hands-On Approach, Orient Blackswan, and some websites)

Q. IoT Protocols

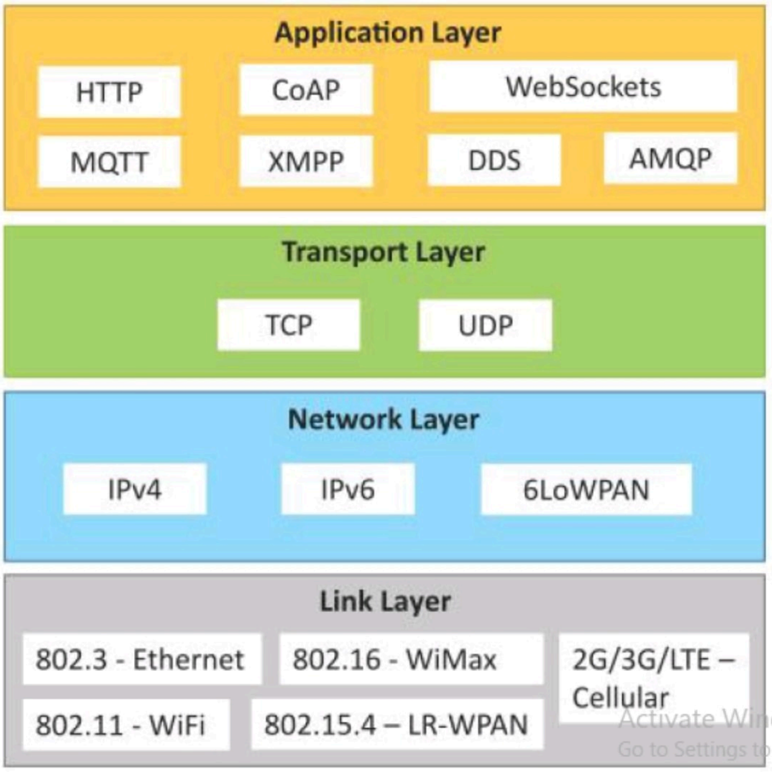


Fig: IoT Protocols

1. Link layer
   - Determines how the data has to be sent through the physical layer (eg: copper wire, radio wave, etc.)

- Determines how the packets are coded and signaled by the hardware device over the medium.
- Its scope is the local network connection to which it is attached.
- Data is exchanged using link layer protocols.
- Some link layer protocols are;
    - 802.3 - Ethernet: Uses coaxial, twisted pair or optical fiber cables.
        - Speed ranges from 10 Mbps to 40 Gbps and higher.
    - 802.11 - Wifi: A collection of wireless LAN (WLAN) technologies.
        - Some of them are 802.11a which operates on 5 GHz frequency band, 802.11b and 802.11g on 2.4 GHz band, 802.11n on both 2.4 and 5 GHz bands, etc
        - Data rate varies from 1 Mbps to 6 Gbps.
    - 802.16 - WiMax: Spans over a wide area.
        - Speed up to 100 Mbps in mobile stations and 1 Gbps in fixed stations.
    - 802.15.4 - LR-WPAN (Low Rate Wireless Personal Area Network):
        - Low cost and low speed communication for power constrained devices.
        - Speed from 40 kbps to 250 kbps.
    - 2G/3G/4G - Mobile Communication: Standards for Cellular Communication methods.
        - 2G - GSM and CDMA
        - 3G - UMTS (Universal Mobile Telecommunications Service)
        - 4G - LTE (Long Term Evolution)
        - Speed ranges from 9.6 kbps (2G) to 100 Mbps (4G)

2. Network Layer (Internet Layer)
    - Sends IP datagrams from the source network to the destination network.
    - Giving addresses to the host.
    - Does packet routing from source to destination across multiple networks.
    - Internet Protocol (IP) Addresses:
        - IPv4: 32 bit address
            - The most commonly deployed addressing scheme
            - Do not guarantee delivery of packets and their integrity, which are handles by upper layer protocols such as TCP
        - IPv6: 128 bit address
            - Came after IPv4 as the IPv4 addresses got exhausted.
        - 6LoWPAN: IPv6 over Low Power Wireless Personal Area Network.
            - For addressing low power low speed devices which have limited processing capability.

3. Transport Layer
    - End to end (process to process) message transfer, independent of underlying network.
    - Major protocols: TCP and UDP
    - TCP (Transmission Control Protocol):
        - Most widely used.

    - Used in web, mail, file transfer etc. (in HTTP, HTTPS, SMTP, FTP)
    - Connection oriented and stateful protocol.
    - Reliable transmission.
    - Does error checking, in-order delivery, flow control, congestion control.
- UDP (User Datagram Protocol);
    - Connectionless, unreliable, stateless protocol.
    - No guaranteed delivery, flow control or congestion control.
    - Useful for time-sensitive applications.

4. Application Layer
    - Defines how applications interface with the lower layer protocols to send the data over the network.
    - HTTP (Hypertext Transfer Protocol)
        - Used by World Wide Web (WWW).
        - Uses request-response method in a client-server model with the help of TCP.
        - An HTTP client can be a browser or an application running on the client (IoT device,mobile phone, etc.)
        - Uses URI (Uniform Resource Identifier) to identify HTTP resources.
        - Common commands include GET, PUT, POST, HEAD, etc.
    - CoAP (Constrained Application Protocol)
        - For machine-to-machine (M2M) applications.
        - Suitable for a constrained environment with constrained devices and constrained networks.
        - Similar to HTTP but uses UDP instead of TCP.
        - Thus a client server model using connectionless datagrams.
        - Common commands include GET, PUT, POST, HEAD, etc.
    - WebSocket:
        - Full duplex communication over a single socket connection.
        - Allows stream of packets back and forth between the client and the server, keeping the TCP connection open.
    - MQTT (Message Query Telemetry Transport)
        - A lightweight messaging protocol based on the **publish-subscribe** model.
        - Client-server model.
        - Clients are IoT devices and servers are called MQTT Brokers.
        - Brokers have several topics to which the clients send (*publish*) messages.
        - Brokers forward the messages to the clients subscribed to these topics.
        - Suitable for a constrained environment with constrained devices and constrained networks.
    - XMPP (Extensible Messaging and Presence Protocol)
        - For real time communication and streaming XML data among IoT and other devices.
        - Used for messaging, gaming, chat applications, voice/video calls, etc.
        - A decentralized, client-server protocol.
        - Supports both client-to-server and server-to-client communication path.

- DDS (Data Distribution Service)
  - A data centric middleware standard for machine-to-machine or device-to-device communication.
  - Uses publish-subscribe model where the publishers (devices that generate data) create topics to which subscribers (devices that want to consume data) can subscribe.
  - Thus publishers distribute data to subscribers.
- AMQP (Advanced Message Queuing Protocol)
  - Supports both point-to-point and publish-subscribe models.
  - Supports routing and queuing of messages.
  - AMQP brokers receive messages from publishers, and copies them into message queues.
  - The messages are either routed by the broker to the subscribers or pulled by the consumers from the queues.

Messaging Protocols

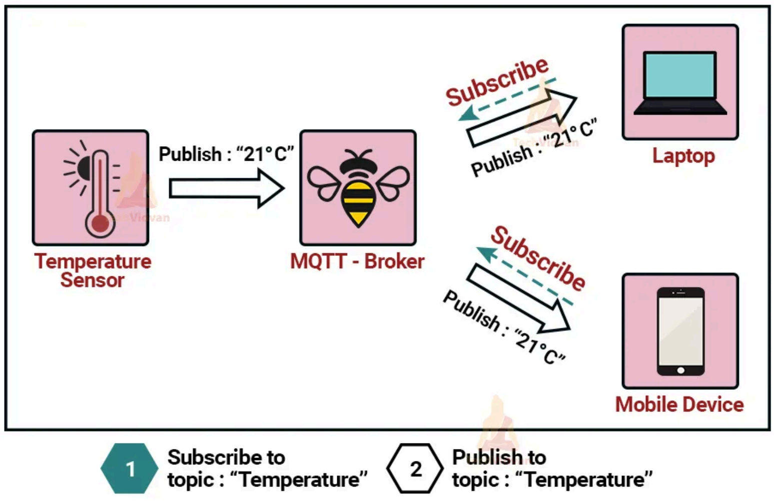
- IoT devices use various kinds of messaging and communication protocols in every layer in order to communicate with each other.
- These messaging protocols are used to transmit device messages (also called *telemetry*) from the IoT devices to the IoT Messaging Hub (or Broker).
- Some common messaging protocols are MQTT and CoAP.

Message Queuing Telemetry Transport (MQTT) Protocol

Features:

- A **publish/subscribe** messaging transport.
- Lightweight: Requires minimal resources so it can be used on small microcontrollers.
- Allows Bi-directional Communications (messaging between device and cloud).
- Security can be enabled which makes it easy to encrypt messages using TLS and authenticate clients using modern authentication protocols, such as OAuth.
- MQTT mainly transmits its data through the TCP/IP protocol.

Architecture:



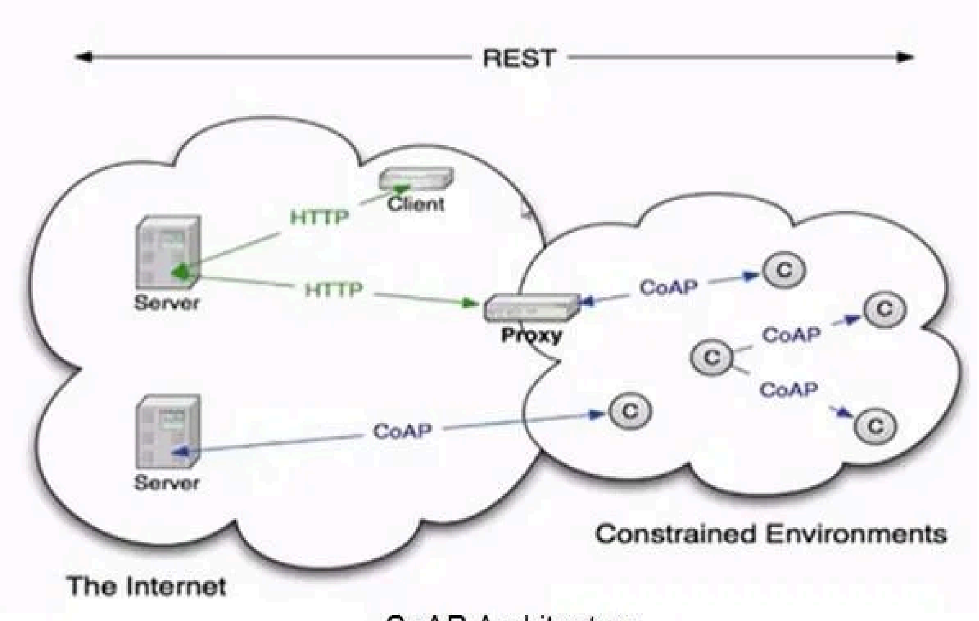
Working:

- There are two types of devices; a **broker** and **clients**.
- All devices communicate through a **broker** or server, which can be installed in PCs, mobile phones, Raspberry Pi, etc. (some famous brokers are HIVEMQ and Mosquito**)**.
- The clients also can range from small microcontrollers with sensors to a full PC.
- The client can either **publish** messages, **subscribe** to specific messages, or do both.
- The broker keeps different categories for messages. These categories are called **topics**. Eg: *temperature of living room, humidity of garden area, etc.*
- When a client sends a message of a particular topic to the broker, the process is called **publish**.
- Other client devices receive information from the broker by just **subscribing** to specific topics.
- When the broker receives a message of a particular topic, it forwards that message to the clients which have subscribed to that topic.
- Three important message types in MQTT are CONNECT, PUBLISH and SUBSCRIBE.

Constrained Application Protocol (CoAP)

- CoAP is an application-layer protocol that is intended for use in low-power, low-bandwidth, **resource-constrained Internet devices**, such as wireless sensor network nodes and IoT nodes.
- Designed to easily translate to HTTP for simplified integration with the web.
- Follows a client-server model in the form of request-response pattern.
- Uses UDP in the transport layer.
- Designed for machine-to-machine (M2M) applications such as smart energy and building automation, supporting constrained devices and networks.
- Enables devices like actuators and sensors to interact over the internet.
- Able to **cooperate with HTTP through simple proxies** and uses the messages like GET, POST, PUT and DELETE verbs similar to HTTP.

Architecture:



CoAP Architecture

- The *Internet* and the *constrained ecosystem* are the two foundational elements of the CoAP protocol architecture.
- In the constrained environment (here, IoT), the nodes (C in the figure) can communicate using CoAP.
- If the server in the Internet can understand CoAP, the nodes can directly communicate with it using CoAP.
- If the server in the Internet can't understand CoAP, it can use HTTP and the proxy device can act as an intermediary between the nodes and the server.

Transport Protocols

Bluetooth Low Energy (BLE)

- BLE is a power-conserving variant of Bluetooth personal area network (PAN) technology, designed for use by Internet-connected machines and appliances, especially in IoT.
- It is a low-power wireless technology that transmits a small amount of data at lower speeds.
- It is used in low bandwidth applications, transfer sensor data, and control devices.
- This is helpful because many IoT nodes use batteries as the power source.
- It has a quicker connection capability.
- It remains in dormant mode until the connection is initiated, hence saving the energy.
- Some common applications for BLE are:
  - Fitness trackers (such as Fitbit, Misfit..etc)
  - Smartwatches (such as the Apple Watch, Moto 360, and Pebble)
  - Beacons (Apple iBeacon, Google Eddystone)
  - Medical devices such as glucose meters, insulin pumps
  - Home automation devices such as door locks, light bulbs, sensors, and others
- Communication between BLE devices can occur in two different ways.
  - Connection-oriented (one to one).
  - based on broadcasting.

Light Fidelity (Li-Fi)

- LiFi is a wireless communication technology which uses the visible light spectrum (as well as the ultraviolet and infrared spectrums, if needed) for high-speed data transfer between devices, terminals, and servers.
- It works by sending information through the LED light bulb, which emits pulsed light to receivers. After that, the receiver collects the sent data or information and understands the transmitted data.
- LiFi provides faster transmission speeds, higher bandwidth, and the ability to work in spaces that are vulnerable to electromagnetic interference, such as airplanes, hospitals, or highly-sensitive industries such as the petroleum industry.
- The drawbacks are the minimal area of connectivity, surrounding light interference and line-of-sight communication.

Assignment 2: Addressing – IPv4, IPv6 - URI