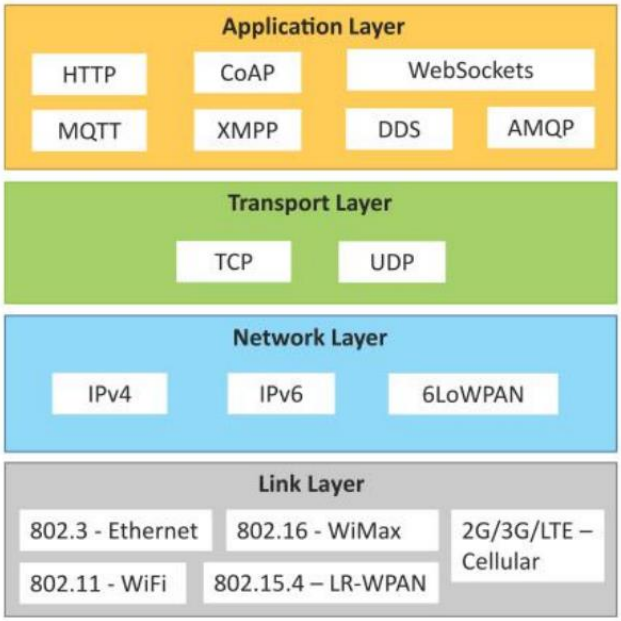
UNIT – 1

PHYSICAL DESIGN OF IoT

ii) IoT Protocol



Link Layer

The Link Layer protocol manages the physical transmission of data over the network's medium and facilitates communication between hosts within the local network connection by encoding and signaling packets through hardware devices.

802.3 Ethernet:

802.3 Ethernet encompasses various wired standards for the link layer, like 10BASE5 Ethernet using coaxial cable and 802.3.i for 10 BASET Ethernet over twisted copper pairs, offering data rates from 10 Mb/s to 40 gigabits per second or more. The shared medium, which can be coaxial cable, twisted pair wire, or optical fiber, facilitates communication among all network devices

802.1- WI-FI:

IEEE 802.11 is a set of wireless Local Area Network (WLAN) standards that define the link layer. Variants like 802.11a, 802.11b, 802.11g, and 802.11ac operate in different frequency bands—

802.11a in the 5 GHz band, and 802.11b, 802.11g, and 802.11ac in the 2.4 GHz and 5 GHz bands.

802.16 WIMAX:

IEEE 802.16, also known as WiMAX, comprises wireless broadband standards with detailed link layer descriptions. WiMAX offers data rates ranging from 1.5 Mb/s to 1 Gb/s, with recent updates delivering speeds of hundreds of megabits per second for mobile stations

802.15.4 LR-WPAN:

IEEE 802.15.4 sets standards for Low Rate Wireless Personal Area Networks (LR-WPAN), forming the basis for protocols like Zigbee, with data rates beginning at 40 kb/s, ideal for affordable, low-speed communication suited to power-constrained devices.

2G / 3G / 4G mobile communications:

These are the different generations of mobile communication standards including second generation (2G including GSM and CDMA). 3rd Generation (3G including UMTS and CDMA2000) and 4th generation 4G including LTE.

Network / internet layer :

The network layer is responsible for sending IP data grams from the source network to the destination network, handling host addressing and packet routing using hierarchical IP addressing schemes like IPv4 or IPv6.

IPv4, the most widely deployed internet protocol, identifies devices on a network using hierarchical addressing schemes, utilizing a 32-bit address format capable of accommodating up to 2^32 addresses. Due to the increasing number of connected devices, IPv4 has been succeeded by IPv6.

IPv6: It is the newest versions of internet protocol and successor to IPv4. IPv6 uses 128-bit address schemes that are lost total of 2 128 are 3.4* 10 38 address.

6LoWPAN brings IPv6 protocol to low-power devices with limited processing capabilities, operating in the 2.4 GHz frequency range and providing data transfer rates of up to 50 kb/s.

Transport Layer:

The Transport layer protocols offer end-to-end message transfer capability regardless of the underlying network, establishing connections with or without handshake acknowledgement. They include functions like error control, segmentation, flow control, and congestion control.

UDP, in contrast to TCP, functions as a connection-less protocol, ideal for time-sensitive applications needing swift data exchange without connection setup. It operates in a transaction-oriented, stateless manner, without ensuring guaranteed delivery, message ordering, or duplicate elimination.

Application layer :

The application layer protocol facilitates the transmission of data between applications by encoding files and encapsulating them within the transport layer protocol, establishing process-to-process connections via ports.

HTTP (Hypertext transfer protocol): is an application layer protocol used for the World Wide Web, allowing clients to send requests to servers using commands like GET, PUT, POST, DELETE, HEAD, TRACE, and OPTIONS, operating on a stateless request-response model, and clients can include web browsers, IoT devices, mobile applications, or other software.

CoAP (Constrained Application Protocol): is designed for machine-to-machine applications in restricted environments, employing a request-response model over UDP, featuring a client-server architecture using connection-less datagrams, and supporting HTTP-like methods such as GET, PUT, and DELETE.

Web Socket: protocol allows bidirectional communication over a single socket connection, letting clients and servers exchange messages seamlessly. It's based on TCP and keeps the connection open for continuous message exchange, catering to various clients like browsers, mobile apps, and IoT devices for real-time

MQTT (Message Queuing Telemetry Transport): is a lightweight protocol using a publish-subscribe model where IoT devices connect to an MQTT broker to publish messages to topics, which are then forwarded to subscribed clients. Its efficiency and simplicity make MQTT well-suited for constrained environments.

XMPP (Extensible Messaging and Presence Protocol): enables real-time communication and XML data streaming between network entities, supporting applications like messaging, presence, gaming, multiparty chat, and voice calls, while facilitating the transmission of small XML data chunks in real time across both client-to-server and server-to-client communication paths.

DDS (Data Distribution Service): facilitates machine-to-machine communication through a publish-subscribe model, where publishers create topics for subscribers to receive data, offering quality of service (QoS) control and configurable reliability for effective communication.

AMQP (Advanced Message Queuing Protocol): enables business messaging with support for routing and queuing through both point-to-point and publish-subscribe models. Brokers in AMQP receive messages from publishers and deliver them to consumers over connections. Publishers transmit messages to exchanges, which then distribute copies to queues.

INTERNET OF THINGS

UNIT – 1

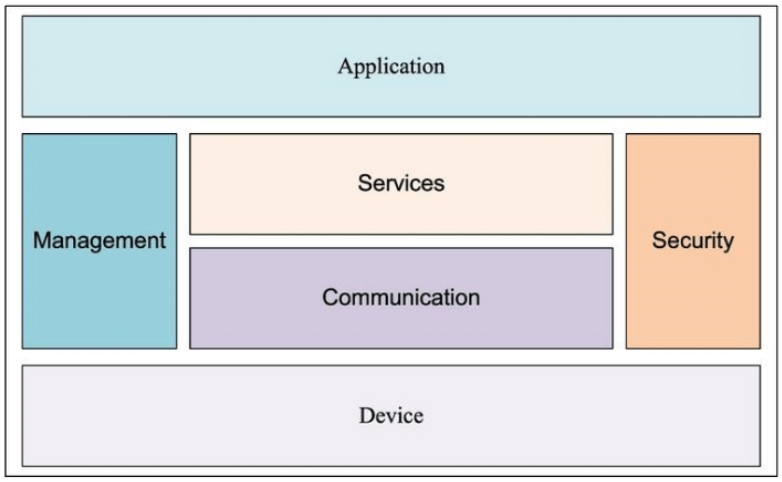
I. LOGICAL DESIGN OF IoT

Logical design of an IoT system refers to an abstract representation of the entities and process without going into low level specification of the implementations.

Logical design of IoT contains the following parameters.

1) IoT functional block diagram
2) IoT Communication models

1) IoT Functional block diagram



An IoT system comprises of a number of functional blocks that provide the system the capabilities for identification, sensing, actuation, communication and Management.

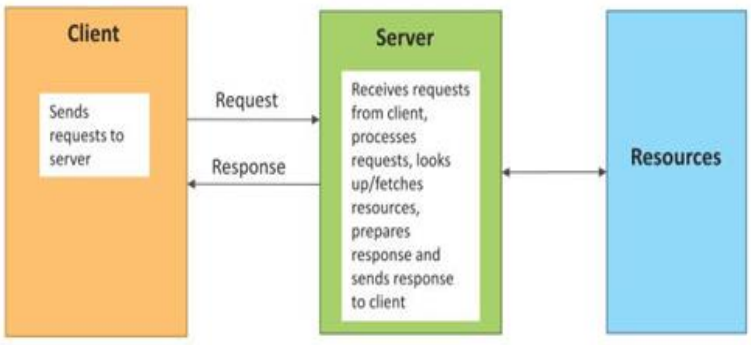
The function blocks are described as follows

- **Devices:** An IoT system comprises of the devices that provide sensing, actuation, monitoring and control function
- **Communication:** communication block handles the communication systems
- **Services:** An IoT system uses various types of IoT services such as services for device monitoring, device control services, data publishing services and services for device ddiscovery.

- **Management:** Functional blocks provide various functions to govern the IoT system
- **Security:** Security functional block security IoT system and by providing functions such as application authorization message and content integrity and data security.
- **Application:** IoT application provides and interface that the user can used to control and monitor various aspects of the IoT system. Application also allow users to view the system status and view or analyze the processed to data.

2) IoT communication models

i) Request-Response communication Prorocol



This communication model consists of two parties:

i) Client ii) server

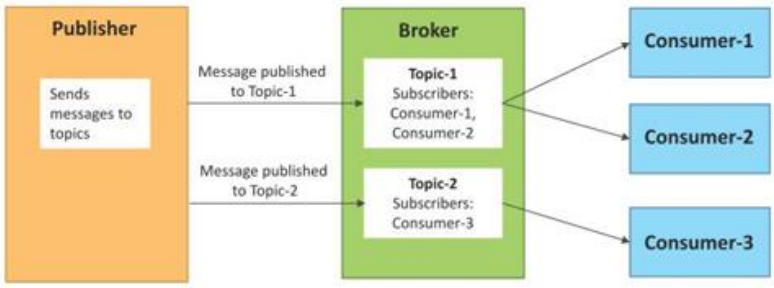
When the client requires any data, it sends a request to the serve r

The server upon receiving the request decides how to respond, fetch the data and sends its response to the client.

Each request-response pair is independent as it is a stateless model.

Ex: Online search engine: when we want to search anything we type a keyword in search bar it fetches all the data relevant to the word we typed and finds all the related websites, images,links on our screen.

ii) publish-subscribe communication Protocol



This communication model consists of three parties:

1) Publisher re data producers or data generators
2) subscriber subscribers are data consumers and
3) Brokers are data mangers

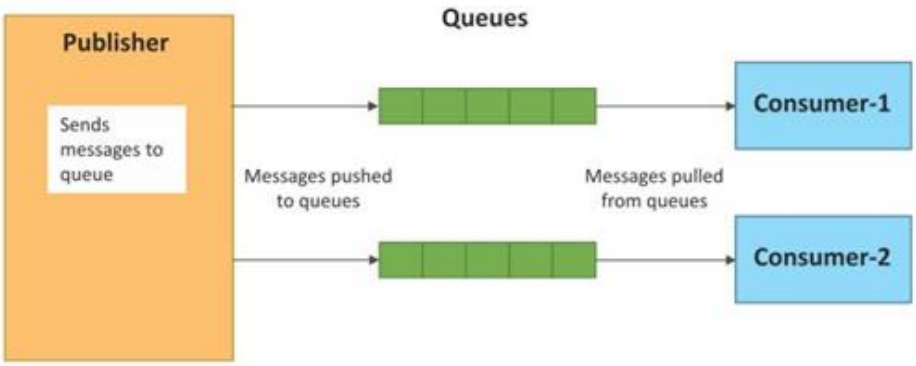
Publishers: They produce data related to various topics let us say sports, fashion, science and technology anything where publishers are not aware of subscribers.

Broker: Forms groups and categorizes of this data according to various topics be it sports, fashion etc borker Identifies the subscribers who has subscribed to the topic 1 then send data related to topic 1 to those subscribers and so on. Publishers and subscribers are connected by the broker, which acts as the connection link.

Subscribers: subscribe various topics according to their interest and brokers/managers provide data to them depending upon their choice of subscription.

Ex: Social media platforms work on this model where we subscribe some channels, like someone's pages or follow someone on social media platform then all the data related to that channel or pages they all appear on our timeline this is how they work because the broker are aware that we have liked that persons page or subscribe that persons channel.

iii) Push-Pull communication model



This communication model consists of two parties:

1) Data producer 2) Data consumers

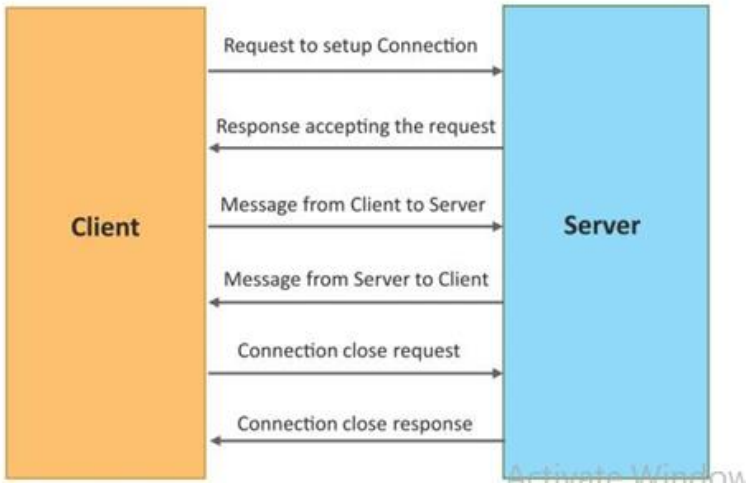
Data producers: generate data related to various topics and push this data into queues (waiting list)

Data consumers: pull data out of these queues

Producers don't require awareness of consumers.

Ex: Social media platforms we scroll down various topics like videos, images, news items them one to our timeline you can visualize as a queue. This queue acts as a buffer or decoupling unit between the data producers and data consumers. If we subscribe some channel then all the data related to that channel comes into timeline from, they are placed in the queue.

iv) <u>Exclusive pair</u>



This communication model consists of two parties:

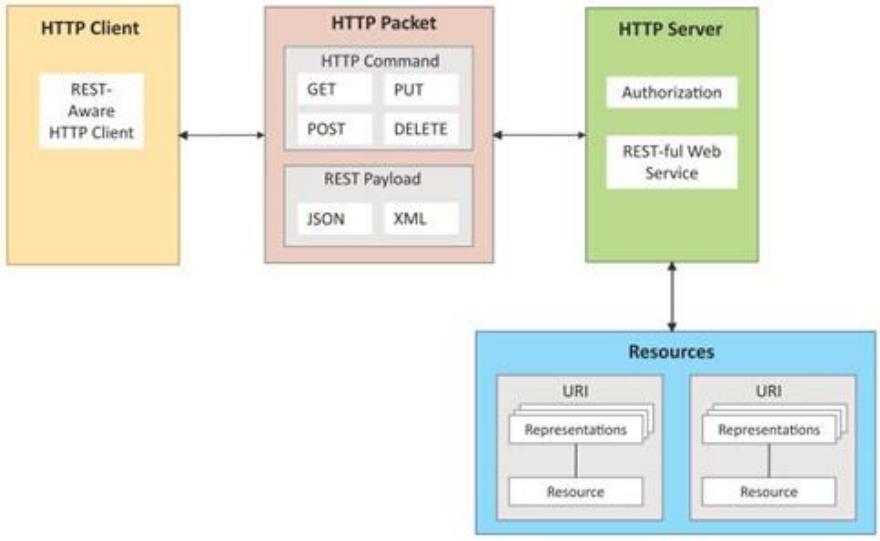
1) client 2) server

Exclusive pair is a bidirectional, fully duplex communication model with persistent connections between client and server, remaining open until the client requests closure. Both client and server can exchange messages after the connection is established. It's a stateful communication model, with the server aware of all open connections.

II. IoT communication APIs

REST-based communication API

REST-based communication API follows representational state transfer principles, focusing on system resources, their states, and addresses for transfer. REST APIs adhere to the request-response communication model, applying architectural constraints to components, connectors, and data elements.



Client server:

Separation of concerns in client-server architecture ensures that the client focuses on user interface while the server handles data storage, enabling independent deployment and updates for each component.

Stateless:

Each request from client to server must contain all the information necessary to understand the request and cannot take advantage of any stored context on the server

Catchable:

By categorizing response data as catchable or non-catchable, the cache allows the reuse of data for similar requests, reducing redundancies and enhancing efficiency and scalability.

Layered system:

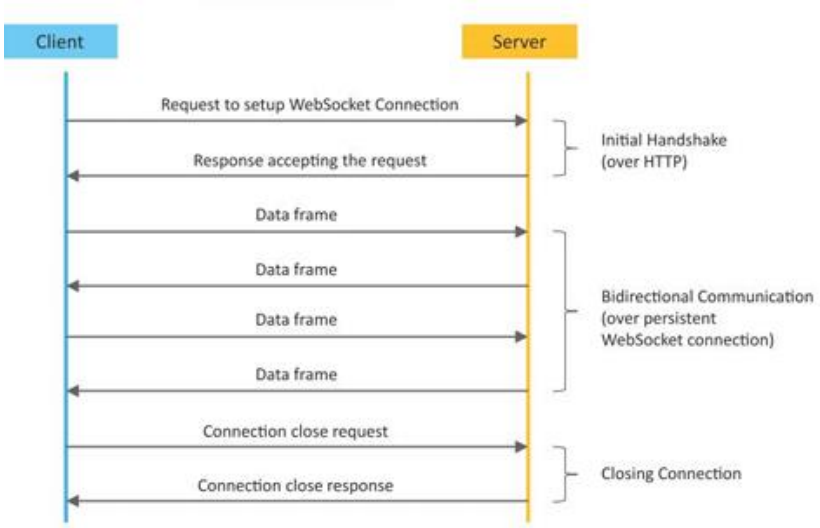
System constraints limit component behavior to interacting only with immediate layers, meaning a client cannot discern if it's connected directly to the end server or an intermediary. By enabling intermediaries to respond to requests rather than the tender server, system scalability can be enhanced

Uniform interface:

The uniform interface constraint mandates consistent communication between client and server, where resources are identified in requests and separate from the representations returned to the client. when the client holds a representation of a resource, it contains all necessary information to update or delete the resource.

Code on demand: Service can provide executable code script for clients to execute in their context.

Web Socket based communication API:



The Web Socket API enables bidirectional, full-duplex communication between client and server without requiring new connections for every message.

It starts with the client sending an HTTP connection setup request, recognized by the server as an upgrade request, followed by a handshake upon protocol support, enabling smooth data transmission. This decreases network traffic and latency by removing connection setup overhead for each message.

III. Difference between REST API and Web socket API

REST API	Web Socket
Stateless	Statefull
Request-response communication model	Full duplex
Each request involves setting up a new TCP connection	Single TCP connection
Header overload	No header overloads
Not suitable for real time applications	suitable for real time

INTERNET OF THINGS

UNIT – 1

IoT enabling Technologies

IoT is enabled by several Technologies

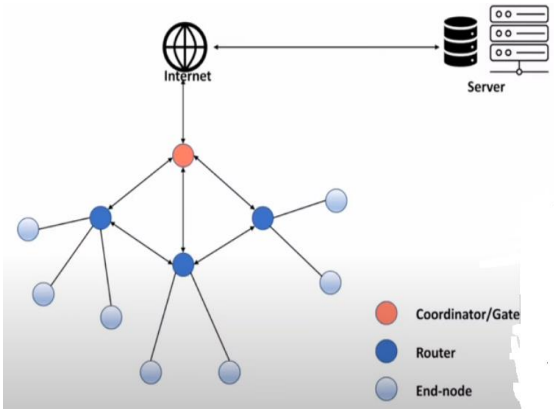
1. Wireless sensor networks
2. Cloud computing
3. Big Data Analytics
4. Embedded system

Wireless Sensor Networks

A WSN consists of devices with sensors for monitoring, including end nodes, routers, and a coordinator, enabling data collection and internet connection.

Wireless Sensor Network (WSN) is comprised of distributed devices equipped with sensors designed to monitor environmental and physical conditions. It consists of end nodes, routers, and a coordinator. End nodes, which can also function as routers, are equipped with multiple sensors.

Routers manage the routing of data packets from end nodes to the coordinator. The coordinator node collects data from all nodes and serves as a gateway, connecting the WSN to the internet.



IoT systems are described as follows

- Weather monitoring utilizes WSNs to collect temperature, humidity, and other environmental data for analysis.
- Indoor air quality monitoring systems employ WSNs to gather data on air quality and detect various gases.
- Soil moisture monitoring systems use WSNs to track moisture levels across different locations.
- Surveillance systems utilize WSNs for collecting motion detection data.
- Smart grids rely on WSNs for monitoring grid conditions at various points.
- Structural health monitoring systems utilize WSNs to monitor structural vibrations at multiple points within a structure.

Cloud Computing: represents a revolutionary computing paradigm where applications and services are delivered via the internet. It involves dynamically allocating computing, networking, and storage resources as needed, offering these resources as metered services in a "pay as you go" model.

Cloud Computing services are offered to user in different forms

Software as a service(SaaS) - Consume

Access of software to client Ex: if you want to use MATLAB then MATLAB service provider will give you access to your computer.

Client does not have any back-end access.

The Cloud Service Provider manages servers, network storage, and application software, shielding users from the underlying architecture. Ex: Goole Workspace, Google Drive etc.

Platform as a service(PaaS) : - Build

Users has no control on underline architecture like storage, OS, server etc.

Service provider gives ability to the user to develop and deploy customer created app using programming languages, tools etc. E: Google App engine (google play store)

Infrastructure as a service(IaaS) : - Host

No worries about underlying physical machine.

Users can deploy operating systems and application of their choice on the virtual resources. Ex: Amazon Web Service , Microsoft Azure.

Big Data Analytics

Big data refers to huge collections of data that are difficult to store, manage, process, and analyze using traditional database tools due to their large volume, velocity, or variety.

Big Data Analytics refers to the methods, tools and applications used to collect, process and drive insights from varied high volume, high velocity data sets involving steps like cleaning, processing, and visualizing the data.

Why Big Data Analytics

Ex: Amazon is learning the requirements of customers as well as absorbing how their competitors doing and based on bid data analytics they are optimizing themselves.

Ex: improving experience of users (Marriott) is having business and hotels here they collect customers experience by facial recognition check-ins.

Ex: Netflix will identify what to suggest to users based on what you are watching on it.

Ex: uber eats helps in business expansion.

Characteristics of data include:

Volume: Big data is a large amount of information that's difficult to handle using traditional methods. Nowadays, IT, industry, and healthcare systems generate tons of data because storing and processing it has become cheaper. We need to find valuable insights within this data to improve business operations and better serve customers.

Velocity: is a critical characteristic of big data, representing the speed at which data is generated and how frequently it changes. Modern IT, industrial, and other systems are producing data at ever-increasing speeds.

Variety: Variety refers to the forms of the data. Big data comes in for different forms such as structured or unstructured data including text data, audio, video and sensor data

Embedded systems

An embedded system is a special type of computer that's made to do certain jobs, using its own built-in hardware and software.

It has parts like microprocessors, memory, networking units, input/output units, display, keyboard, and storage like flash memory.

Some of these systems also come with special processors like digital signal processors (DSP) and graphic processors.

Ex: Washing machine.

Drones.

Tracking systems.

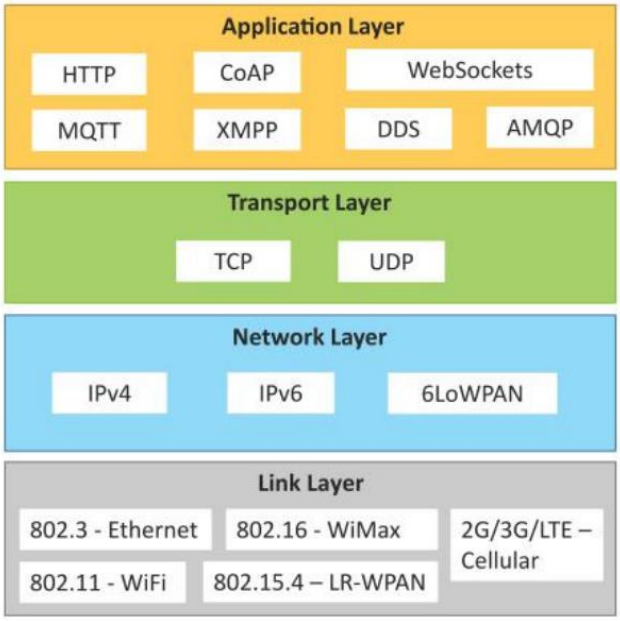
Electric vehicle charging stations.

Wearable devices etc.,

UNIT – 1

PHYSICAL DESIGN OF IoT

ii) IoT Protocol



Link Layer

The Link Layer protocol manages the physical transmission of data over the network's medium and facilitates communication between hosts within the local network connection by encoding and signaling packets through hardware devices.

802.3 Ethernet:

802.3 Ethernet encompasses various wired standards for the link layer, like 10BASE5 Ethernet using coaxial cable and 802.3.i for 10 BASET Ethernet over twisted copper pairs, offering data rates from 10 Mb/s to 40 gigabits per second or more. The shared medium, which can be coaxial cable, twisted pair wire, or optical fiber, facilitates communication among all network devices

802.1- WI-FI:

IEEE 802.11 is a set of wireless Local Area Network (WLAN) standards that define the link layer. Variants like 802.11a, 802.11b, 802.11g, and 802.11ac operate in different frequency bands—

802.11a in the 5 GHz band, and 802.11b, 802.11g, and 802.11ac in the 2.4 GHz and 5 GHz bands.

802.16 WIMAX:

IEEE 802.16, also known as WiMAX, comprises wireless broadband standards with detailed link layer descriptions. WiMAX offers data rates ranging from 1.5 Mb/s to 1 Gb/s, with recent updates delivering speeds of hundreds of megabits per second for mobile stations

802.15.4 LR-WPAN:

IEEE 802.15.4 sets standards for Low Rate Wireless Personal Area Networks (LR-WPAN), forming the basis for protocols like Zigbee, with data rates beginning at 40 kb/s, ideal for affordable, low-speed communication suited to power-constrained devices.

2G / 3G / 4G mobile communications:

These are the different generations of mobile communication standards including second generation (2G including GSM and CDMA). 3rd Generation (3G including UMTS and CDMA2000) and 4th generation 4G including LTE.

Network / internet layer :

The network layer is responsible for sending IP data grams from the source network to the destination network, handling host addressing and packet routing using hierarchical IP addressing schemes like IPv4 or IPv6.

IPv4, the most widely deployed internet protocol, identifies devices on a network using hierarchical addressing schemes, utilizing a 32-bit address format capable of accommodating up to 2^32 addresses. Due to the increasing number of connected devices, IPv4 has been succeeded by IPv6.

IPv6: It is the newest versions of internet protocol and successor to IPv4. IPv6 uses 128-bit address schemes that are lost total of 2 128 are 3.4* 10 38 address.

6LoWPAN brings IPv6 protocol to low-power devices with limited processing capabilities, operating in the 2.4 GHz frequency range and providing data transfer rates of up to 50 kb/s.

Transport Layer:

The Transport layer protocols offer end-to-end message transfer capability regardless of the underlying network, establishing connections with or without handshake acknowledgement. They include functions like error control, segmentation, flow control, and congestion control.

UDP, in contrast to TCP, functions as a connection-less protocol, ideal for time-sensitive applications needing swift data exchange without connection setup. It operates in a transaction-oriented, stateless manner, without ensuring guaranteed delivery, message ordering, or duplicate elimination.

Application layer :

The application layer protocol facilitates the transmission of data between applications by encoding files and encapsulating them within the transport layer protocol, establishing process-to-process connections via ports.

HTTP (Hypertext transfer protocol): is an application layer protocol used for the World Wide Web, allowing clients to send requests to servers using commands like GET, PUT, POST, DELETE, HEAD, TRACE, and OPTIONS, operating on a stateless request-response model, and clients can include web browsers, IoT devices, mobile applications, or other software.

CoAP (Constrained Application Protocol): is designed for machine-to-machine applications in restricted environments, employing a request-response model over UDP, featuring a client-server architecture using connection-less datagrams, and supporting HTTP-like methods such as GET, PUT, and DELETE.

Web Socket: protocol allows bidirectional communication over a single socket connection, letting clients and servers exchange messages seamlessly. It's based on TCP and keeps the connection open for continuous message exchange, catering to various clients like browsers, mobile apps, and IoT devices for real-time

MQTT (Message Queuing Telemetry Transport): is a lightweight protocol using a publish-subscribe model where IoT devices connect to an MQTT broker to publish messages to topics, which are then forwarded to subscribed clients. Its efficiency and simplicity make MQTT well-suited for constrained environments.

XMPP (Extensible Messaging and Presence Protocol): enables real-time communication and XML data streaming between network entities, supporting applications like messaging, presence, gaming, multiparty chat, and voice calls, while facilitating the transmission of small XML data chunks in real time across both client-to-server and server-to-client communication paths.

DDS (Data Distribution Service): facilitates machine-to-machine communication through a publish-subscribe model, where publishers create topics for subscribers to receive data, offering quality of service (QoS) control and configurable reliability for effective communication.

AMQP (Advanced Message Queuing Protocol): enables business messaging with support for routing and queuing through both point-to-point and publish-subscribe models. Brokers in AMQP receive messages from publishers and deliver them to consumers over connections. Publishers transmit messages to exchanges, which then distribute copies to queues.

INTERNET OF THINGS

UNIT – 1

PHYSICAL DESIGN OF IoT

- Physical design is more than creating products. It includes things/devices and protocols
- Node devices perform remote sensing, actuating and mentoring work. The protocols are used to establish communication between the node devices and servers.
- It is about forming overall intelligent system.
- In physical design one understands the most effective method of storing and accessing the objects.

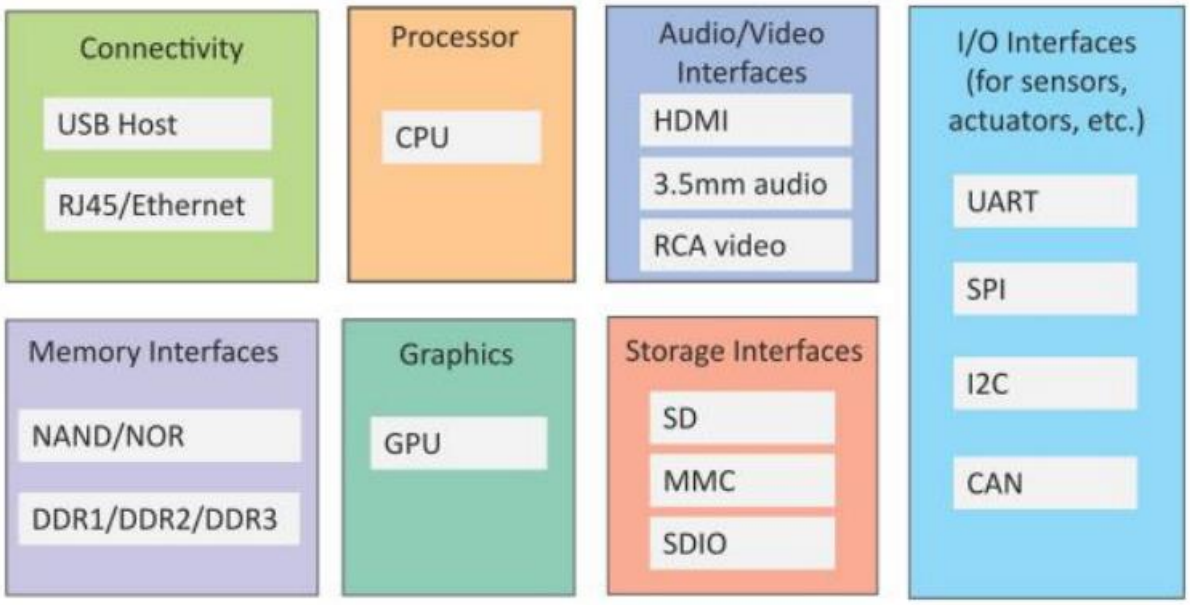
I. Things of IoT

- In IoT, "Things" typically refers to devices with unique identities capable of remote sensing, Actuation, and monitoring. These devices exchange data with other connected devices and applications, either directly or indirectly, and collect data from other devices. They can process data locally or transmit it to centralized servers or cloud-based applications for further processing, considering constraints like memory, processing power, communication speed, and deadlines within the IoT infrastructure.
- IoT devices can connect to other devices through interfaces like sensors, the internet, memory, and audio-video connections, and they collect various data types such as temperature, humidity, and light intensity from onboard or attached sensors.

- Things in IoT are pertain to devices that are used for actuating, Monitoring and remote sensing capabilities. It is used for following functions:

    1. Collect data (sensing physical data)
    2. Exchange data (Things to IoT cloud)
    3. Perform tasks based on programs/algorithm (Actuator)

II. GENERIC BLOCK DIAGRAM OF IoT DEVICE



Connectivity: connectivity should be given to things with IoT server and for that one can have connectivity protocols based on USB or EHERNET

Processor: focus is on CPU generally in embedded system will have microprocessor and microcontroller that is used to optimize our applications by which one can decide operational speed of given application.

Audio/Video Interface: input and output are provided with audio interface where you can have HDMI RAC for video for audio you can have 3.5mm.

I/O interfaces: consists of sensors and actuators for communication we have serial communication protocol like UART, SPI, I2C and CAN.

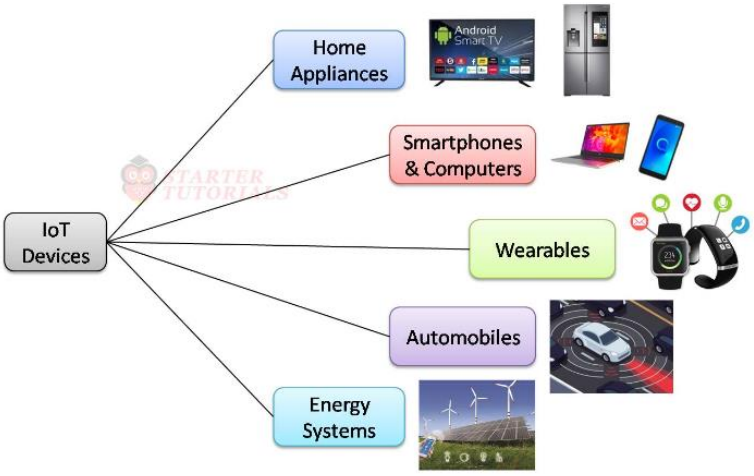
Storage Interface: where storage device like SD, MMC, SDIO are available.

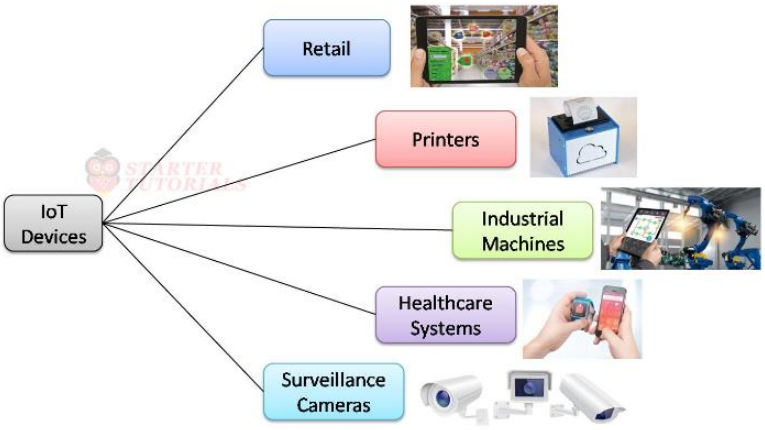
Graphics: For excellent performance of picturization graphics is required.

Memory Interface: memory interface like DDR1/DDR2/DDR3 or we can use NAND/NOR gates.

III. IOT DEVICES

IoT devices can also be varied types, for instance, wearable sensors, smart watches, LED light automobiles and industrial machines





INTERNET OF THINGS

UNIT – 1

INTRODUCTION TO IoT

I. DEFINITION AND CHARACTERISTICS OF IoT

DEFINITION

A dynamic global network infrastructure with self–configuring based on standard and interoperable communication protocols where physical and virtual "things" have identified, physical attributes, and virtual personalities and use intelligent interfaces, often communicate data associated with users and their environment

II. CHARACTERISTICS OF IoT

Dynamic and self-Adapting: IoT devices and systems may have the capability to dynamically adapt with the changing contexts and take actions based on their operating condition. Ex: Surveillance cameras can adapt their modes based on whether it is day or night.

Self – Configuring: IoT devices may have self-Configuring capability allowing a large number of devices to work together to provide certain functionality. Ex: In a weather monitoring system, sensors placed across a large area use natural energy sources and receive remote updates or instructions from a central server, reducing the need for manual visits to update them.

Interoperable communication protocols: IoT devices can communicate with each other and with infrastructure using various interoperable communication protocols.

Unique Identity: Each IoT device possesses a distinct identity and identifier, such as an IP address or URI. Additionally, IoT systems feature intelligent interfaces that adjust according to context, facilitating communication with users and environmental factors.

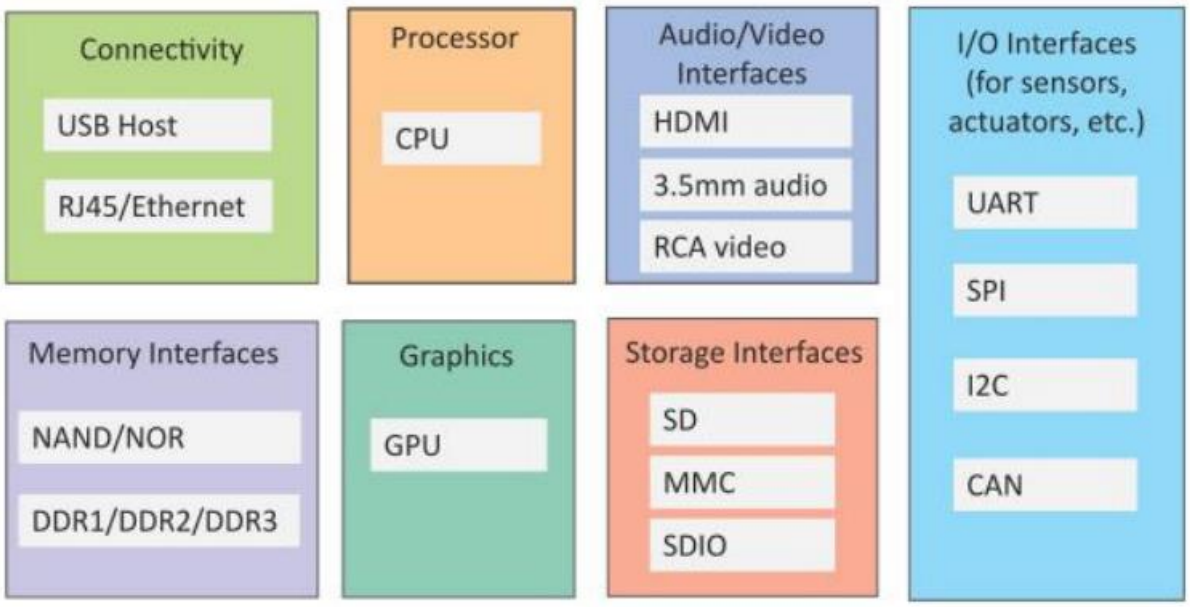
Integrated into information network: IoT devices are usually integrated into the information network that allows them to communicate and exchange data with other devices and systems.

Ex: When a weather monitoring node shares its abilities with another connected node, they can exchange data, enhancing IoT systems by combining device intelligence with infrastructure, which allows the aggregation and analysis of data from numerous weather monitoring nodes to predict the weather.

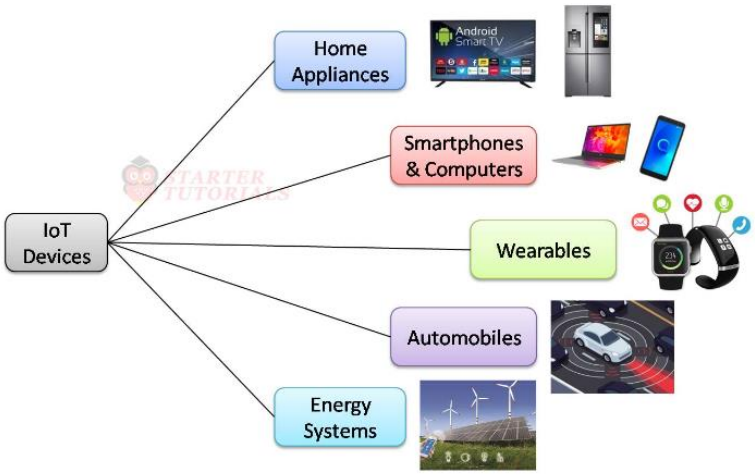
III. PHYSICAL DESIGN OF IoT

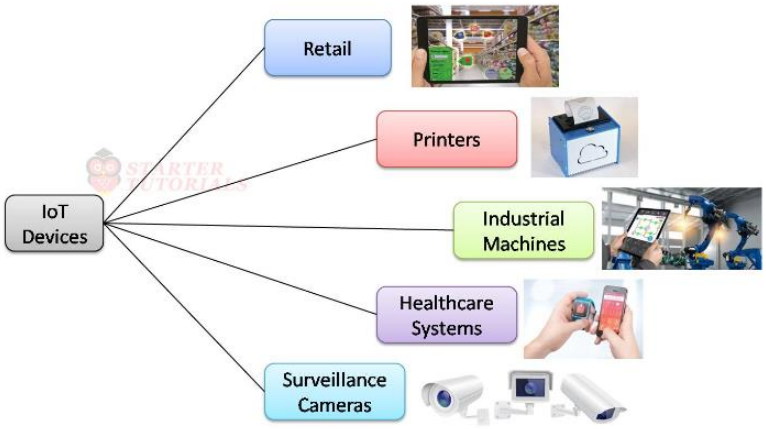
i) Things of IoT

- In IoT, "Things" typically refers to devices with unique identities capable of remote sensing, Actuation, and monitoring. These devices exchange data with other connected devices and applications, either directly or indirectly, and collect data from other devices. They can process data locally or transmit it to centralized servers or cloud-based applications for further processing, considering constraints like memory, processing power, communication speed, and deadlines within the IoT infrastructure.
- IoT devices can connect to other devices through interfaces like sensors, the internet, memory, and audio-video connections, and they collect various data types such as temperature, humidity, and light intensity from onboard or attached sensors.



- IoT devices can also be varied types, for instance, wearable sensors, smart watches, LED light automobiles and industrial machines





Application Layer Protocols

Application Layer

- Defines how the applications interface with lower layer protocols to send data over thenetwork.
- Enables process-to-process communication using port numbers(for example port 80for HTTP, Port 22 for SSH,etc.)

Protocols:

- HTTP (Hyper Text transfer protocol)
- COAP (Constrained Application Protocol)
- Web Socket (Allows Full duplex communication
- MQTT (Message Queue Telemetry Transport
- XMPP (Extensible Messaging Presence Protocol)
- DDS (Data Distribution Service)
- AMQP (Advanced Message Queuing Protocol)

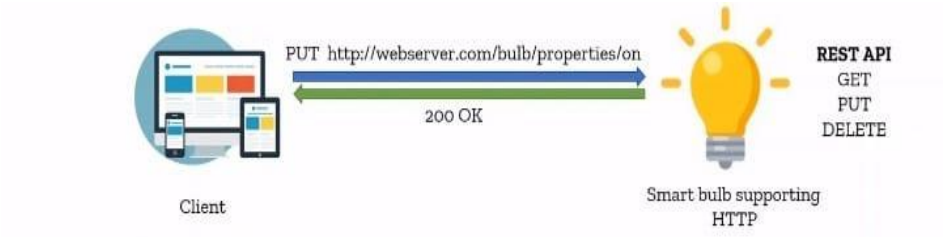
HTTP: The Hypertext Transfer Protocol

- Most widely used protocol of Application layer. Allows Half –Duplex communication.
- Stateless Protocol
- Server will not remember previous request.
- Each HTTP Request is independent of other.
- Follows Request-response model where a client sends requests to server using HTTP commands

- HTTP runs over TCP/IP to ensure packet delivery guarantee. Because their should be communication between client and server to send message for this packet delivery guarantee is required for this reason we use TCP/IP.
- HTTP included commands such as GET, PUT, POST, DELETE, HEAD, TRACE, OPTIONS, etc.

How HTTP works in IoT.

- HTTP Implementation based on REST (Representational State Transfer) is a Standard/design to communicate between systems on web where as HTTP is a implementation based on REST standard.
- If any architecture follows REST standard the called as RESTful by default



Web Socket Protocol:

- Allows full –Duplex communication over a single socket connection for sending messages betweenclient and server.

- Based on TCP.
- Web Socket allows streams of messages to be sent back and forth between the client and server whilekeeping TCP connection open.
- Web Socket works on all modern browsers viz., Firefox, Chrome etc..

How Web Socket Works in IoT

- Useful for real time communication without any delay in an IoT applications.
- Faster
- Not Preferable for battery operated devices.

MQTT (Message Queue Telemetry Transport)

- Message Queue Telemetry Transport is light weight messaging protocol based on **publish-subscribemodel.**
- Mostly widely used in IoT Applications, runs on **TCP/IP**
- Uses client server architecture, where the client(IoT device) connects to the server(also called asMQTT Broker) and publishes messages to topics on the server.

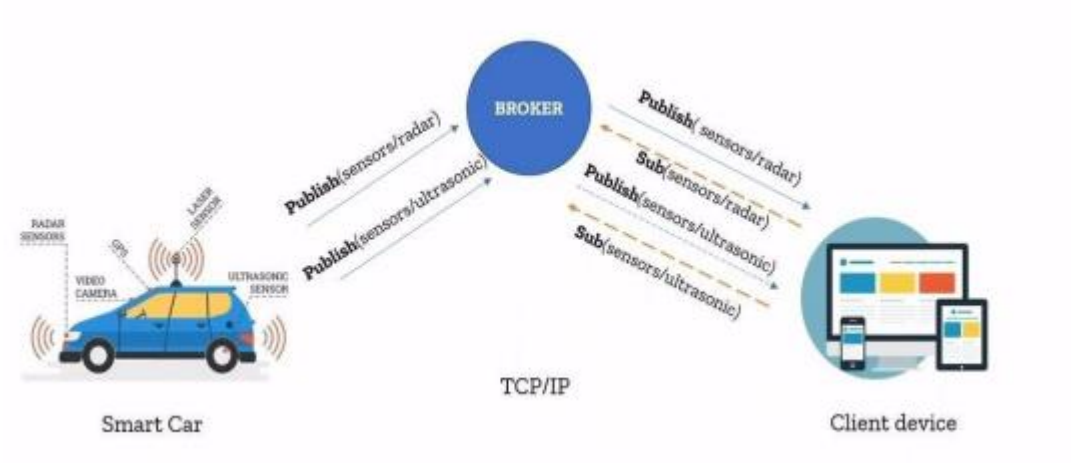
- The Broker forwards the messages to the clients subscribed to the topics.
- No **direct connection** between two clients/nodes, its always through Broker/Server
- Well suited for constrained environment where the devices have limited processing ,memoryresources ,low power and low bandwidth.
- Offers three levels of QoS

  --QoS 0: Fire & Forget(No Guarantee of delivery)

  --QoS 1: Delivery at least Once

  --QoS 2: Delivery exactly Once

How MQTT Works in IoT



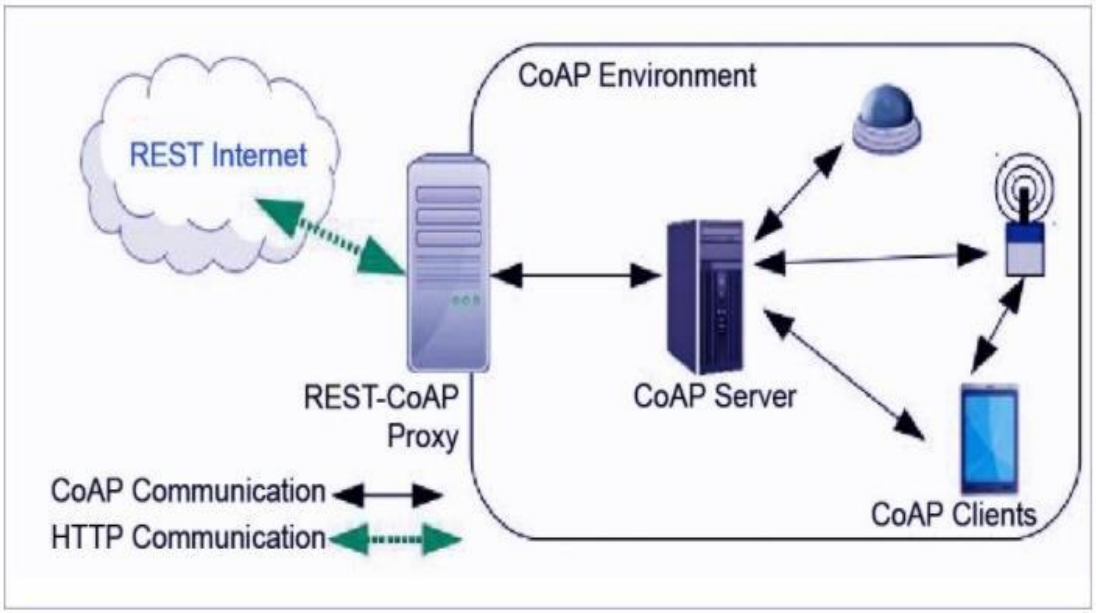
A smart car equipped with various sensors such as radar, laser, and ultrasonic gathers data and sends it to a broker with specific topics indicating the source, such as "publish/sensor/radar" for radar sensor data. Client devices subscribe to the broker using corresponding topics like "sub/sensor/radar" to receive data from specific sensors. The broker then delivers the data to the respective clients awaiting information from the designated sensors.

CoAP (Constrained Application Protocol):

- Designed for machine-to-machine (M2M) applications such as smart energy and building automation.
- The Constrained Application Protocol (CoAP) is a specialized web transfer

protocol for use with constrained nodes and constrained networks.(Low Power,low bandwidth,etc)

- Request-response model
- Runs on top of UDP
- CoAP uses Client-Server architecture where clients communicate with server using connection-less datagrams
- Direct Connection between Client & Client is possible.
- CoAP supports commands such as GET, PUT, POST, DELETE

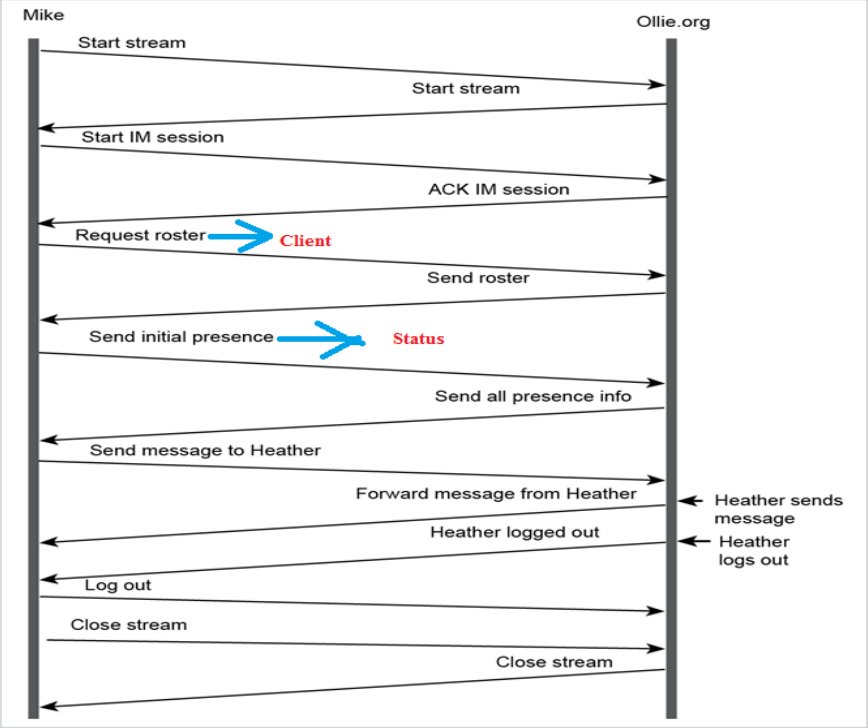


Implements **DTLS (Datagram Transport Layer security)** for secure message exchange inTransport layer

I. APPLICATION LAYER PROTOCOLS

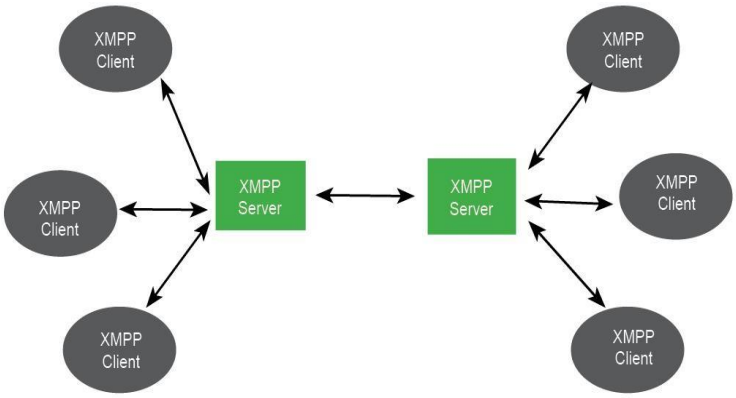
XMPP (Extensible Messaging and Presence Protocol)

- Formerly known as Jabber, is acommunications protocol based on XML (Extensible Markup Language)
  - **X :** It means eXtensible. XMPP is a open source project which can be changed or extended accordingto the need.
  - **M :** XMPP is designed for sending messages in real time. It has very efficient push mechanismcompared to other protocols.
  - **P :** It determines whether you are online/offline/busy. It indicates the state.
  - **P :** XMPP is a protocol, that is, a set of standards that allow systems to communicate with each other.



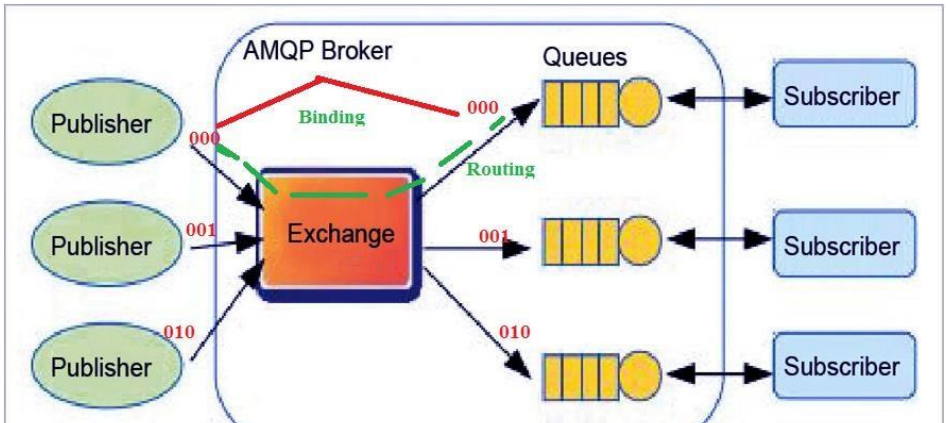
- Wide range of applications including messaging, presence, data syndication(aggregated collection),gaming, group video calls
- Supports both client-to -server & Server -to - server communication paths.

- Used in chat apps.
- XMPP uses Port 5222 for the client to server (C2S) communication and utilizes Port 5269 for server to server (S2S) communication.



AMQP: Advanced Message Queuing protocol

- Open application layer for business messaging.
- Supports both Point-to-Point & Publisher/subscriber models, routing and Queuing.
- Messages are either delivered by the broker to the consumers which have subscribed to the queues orthe consumers can pull the messages from the queues.
- The AMQP protocol enables patron programs to talk to the dealer and engage with the AMQP model. This version has the following three additives, which might link into processing chains in the server to create the favored capabilities.
- **Exchange:** Receives messages from publisher primarily based programs and routes them to 'messagequeues'.
- **Message Queue:** Stores messages until they may thoroughly process via the eating client software.
- **Binding:** States the connection between the message queue and the change.



The publisher will transmit data to an exchange, which will then direct it to one of several queues based on the exchange type. When a subscriber expresses interest in specific data, the relevant queue will send that data to the corresponding subscriber.

Binding: it is link between exchange and queue. The type of exchange are
Direct, ind

In a Direct Exchange, when the binding key matches the routing key, the exchange will exclusively deliver the data to the corresponding queue.

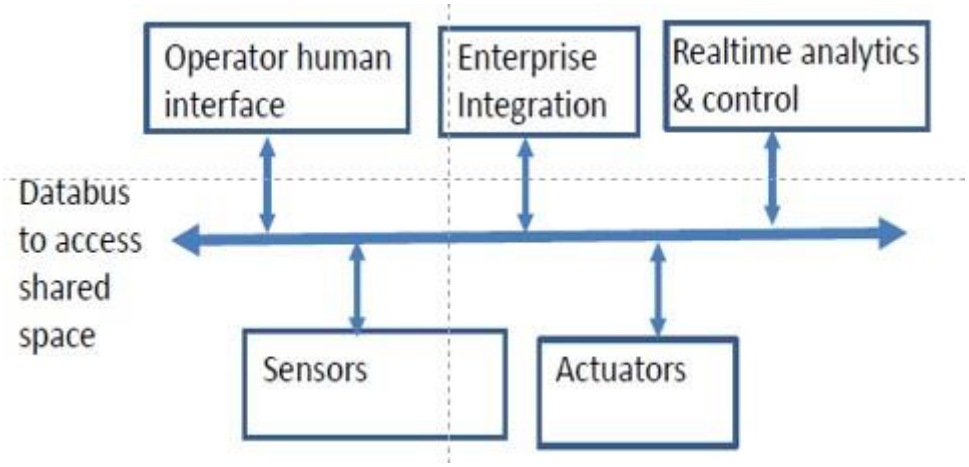
Fan-out: The exchange will route any incoming message to all queues bound to it.

Topic: It is trying to match routing key and routing pattern(which is there on the topic). if matches route the data if not don't route the data.

Header: Focus on header attributes of the message based on it route the data or not.

DDS (Data Distribution Service):

- Is a data centric middleware standard for Device to Device or machine to machine communication.
- Publish-Subscribe model.
- Publishers (IoT device which generates data) create topics to which subscribers(device whichreceives data) can subscribe.
- Provides Quality of service (Qos) control, configurable & reliability
- DDS makes use of multi casting to convey high-quality QoS to applications.

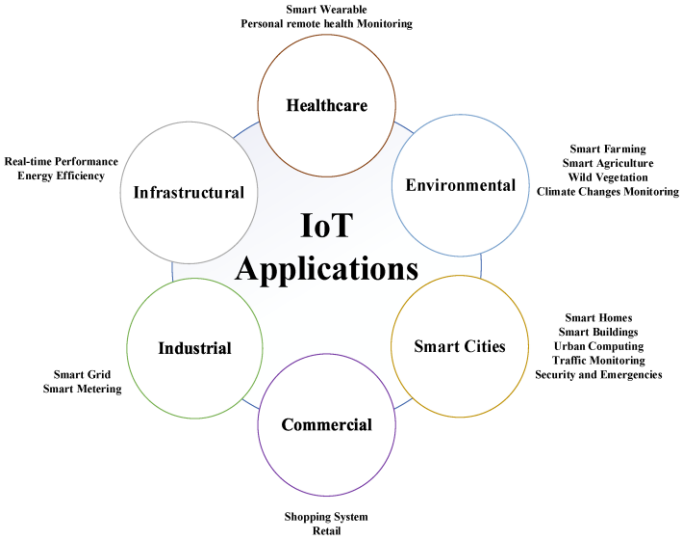


- Above data bus are applications and below are sensors and actuators
- It is the service that distributes data. The data hear means device data can include kernel data, operating system used, base-band version etc.. (Example an android phone talking to google assistant like siri)

Applications of DDS are.

- Military system
- Medical imaging
- Asset tracking
- Hospital integration.

Applications of IoT:



- Smart Homes
- Smart Cities
- Environment
- Energy
- Retail
- Logistics
- Agriculture
- Industry
- Health & Lifestyle