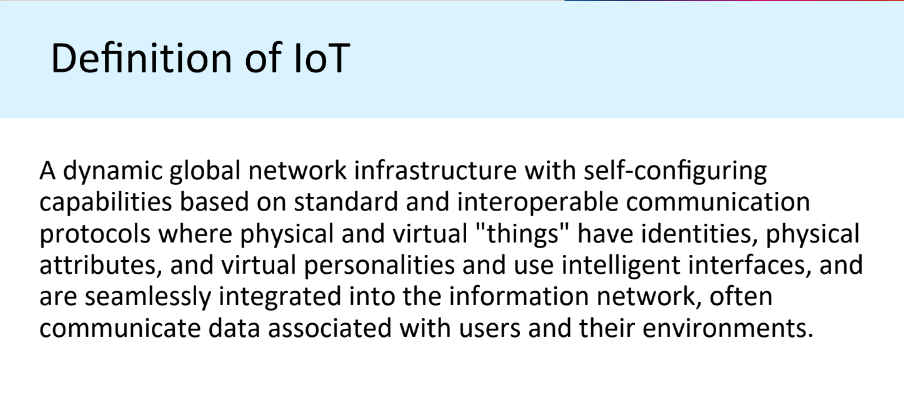
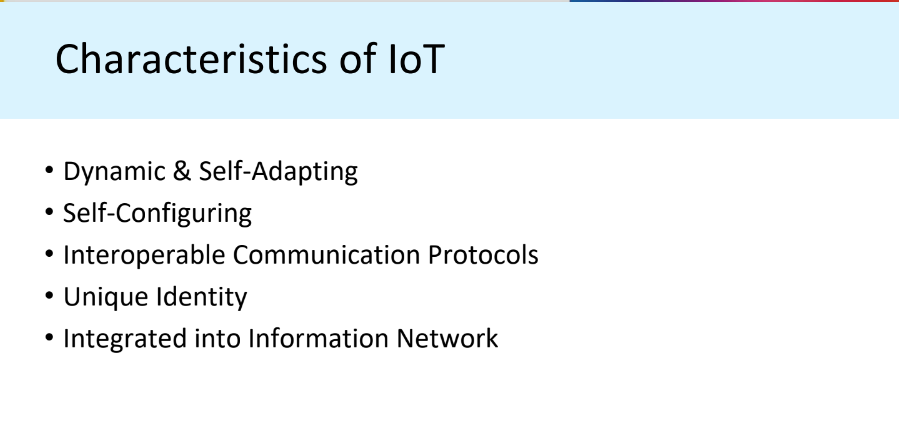
**Module 1:** Introduction to IoT

**Introduction to IoT- Defining IoT**



**Characteristics of IoT**



**Conceptual Framework of IoT**

Physical design of IoT, Logical

design of IoT, Functional blocks of IoT, Brief review of

applications of IoT.

**Module 2**:IoT Architecture

**Drivers Behind New Network Architectures**:

Scale, Security, Constrained Devices and Networks, Data, Legacy Device Support.

**Architecture**: The IoT World Forum (IoTWF) Standardized Architecture: Layer 1-7,

IT and OT Responsibilities in the IoT Reference Model,

Additional IoT Reference Models

A Simplified IoT Architecture The Core IoT Functional Stack: Layer 1-3,

Analytics Versus Control Applications, Data Versus Network

Analytics Data Analytics Versus Business Benefits, Smart Services,

**IoT Data Management, and Compute Stack**: Fog Computing, Edge Computing, The Hierarchy of Edge, Fog, and Cloud

**Module 3:** Principles of Connected Devices and Protocols in IoT

* Sensors: Temperature, Humidity, level, flow, etc.
* A sensor: It senses
* More specifically, a sensor measures some physical quantity and converts that measurement reading into a digital representation.
* That digital representation is typically passed to another device for transformation into useful data that can be consumed by intelligent devices or humans
* Sensors are not limited to human-like sensory data.
* They are able to provide an extremely wide spectrum of rich and diverse measurement data with far greater precision than human senses

### **Classification of sensors**

1. **Passive –** requiring an external excitation to produce an electrical output
2. **Active –** producing a voltage output without any external excitation

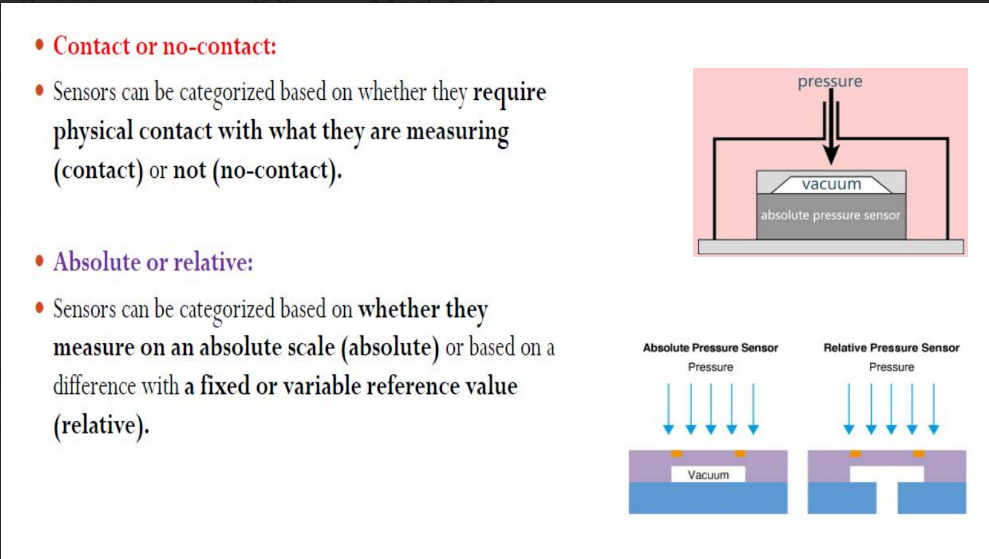
A Thermocouple is an Active Sensor as it does not require any external power supply to operate.

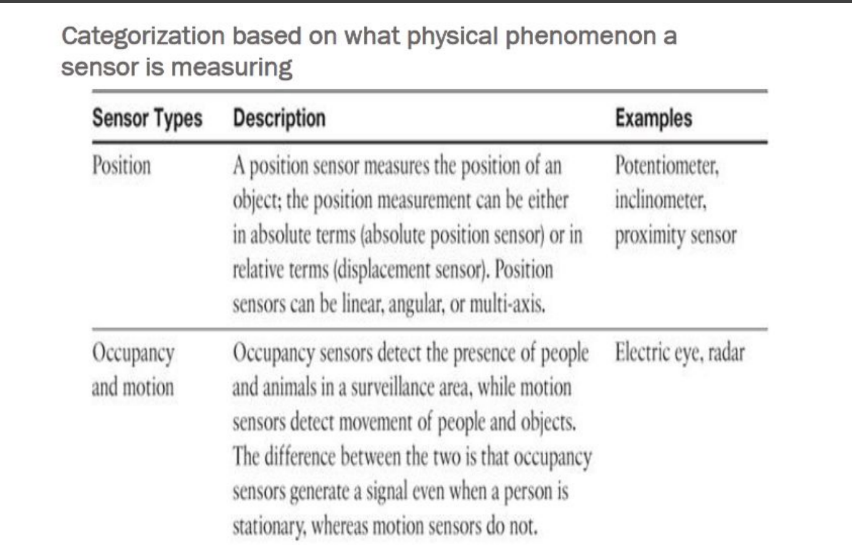
Another example of an Active sensor is a piezoelectric sensor.

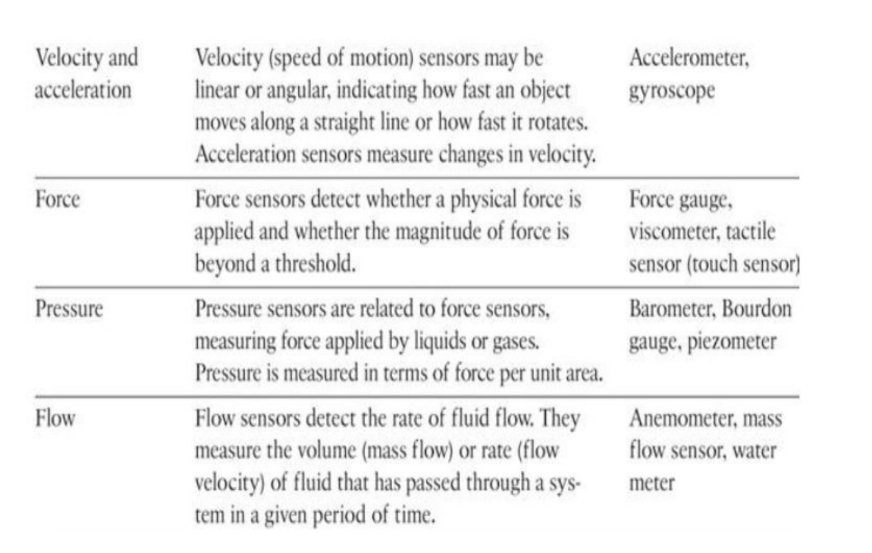
A Resistance Temperature Detector (RTD) is a Passive Sensor. It is a device that’s resistance will change with a change in temperature. To take advantage of this change in resistance, an external supply, or an excitation circuit is required to produce a change in voltage.

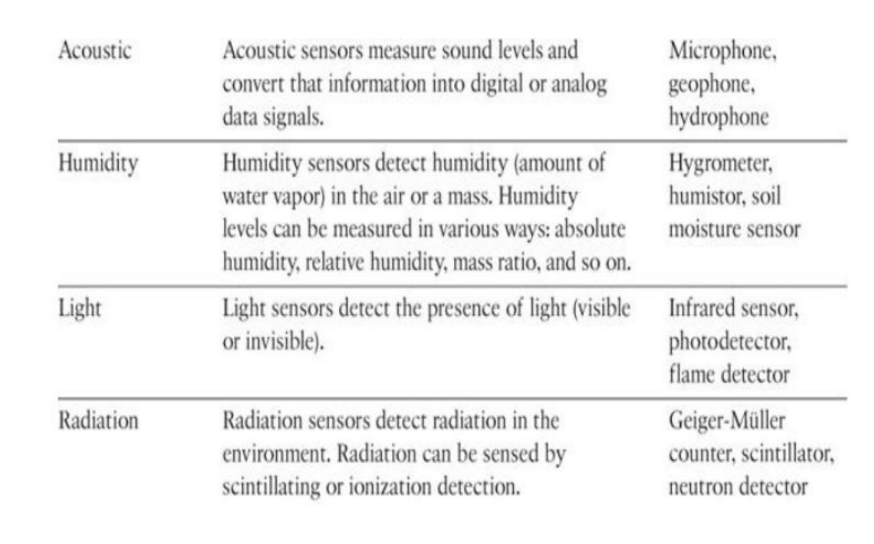
**Invasive or non-invasive**

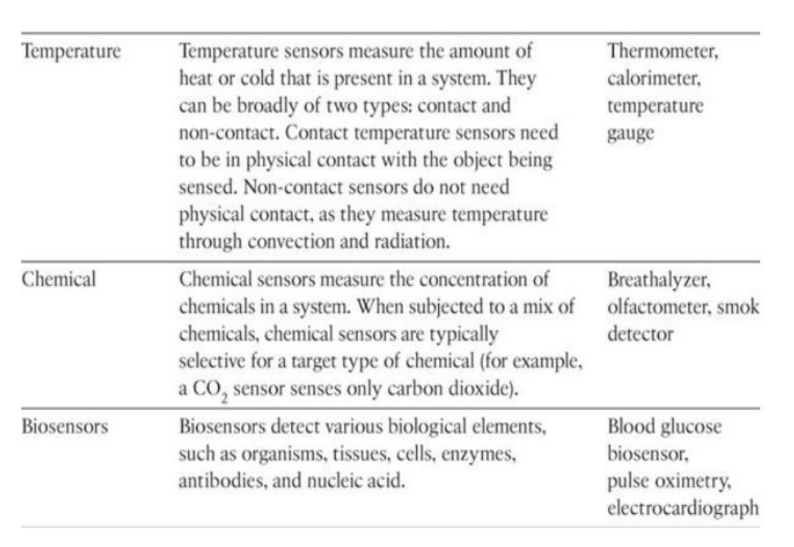
Sensors can be categorized based on whether a sensor is part of the environment it is measuring (invasive) or External to it (non-invasive).





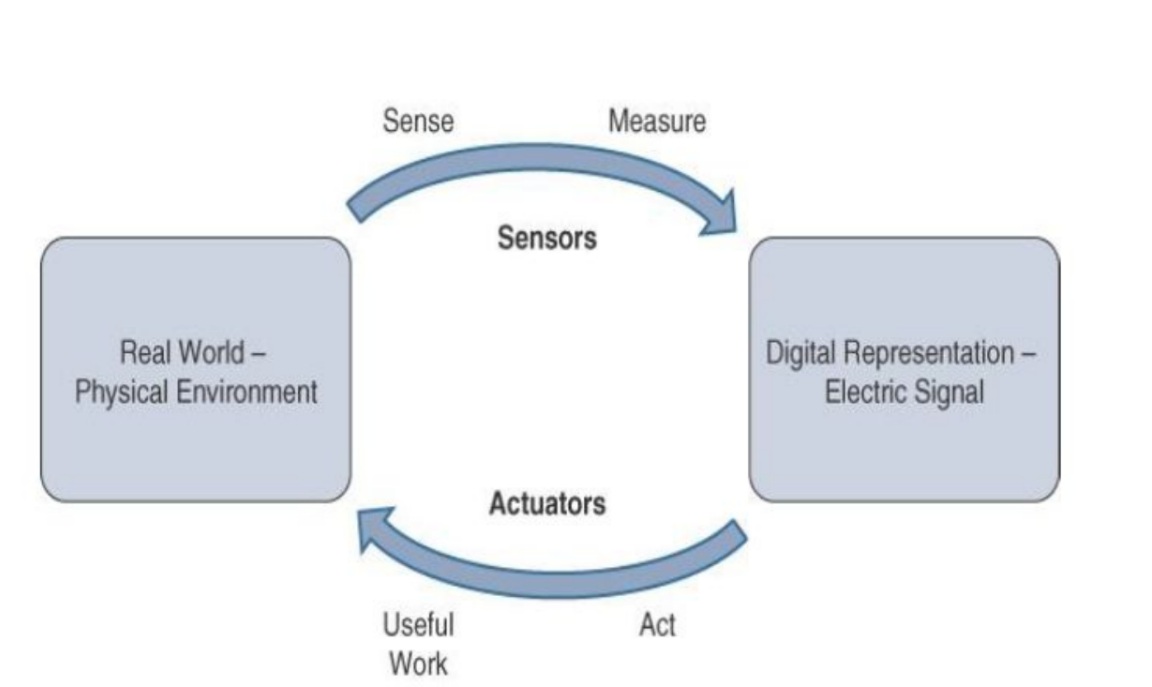




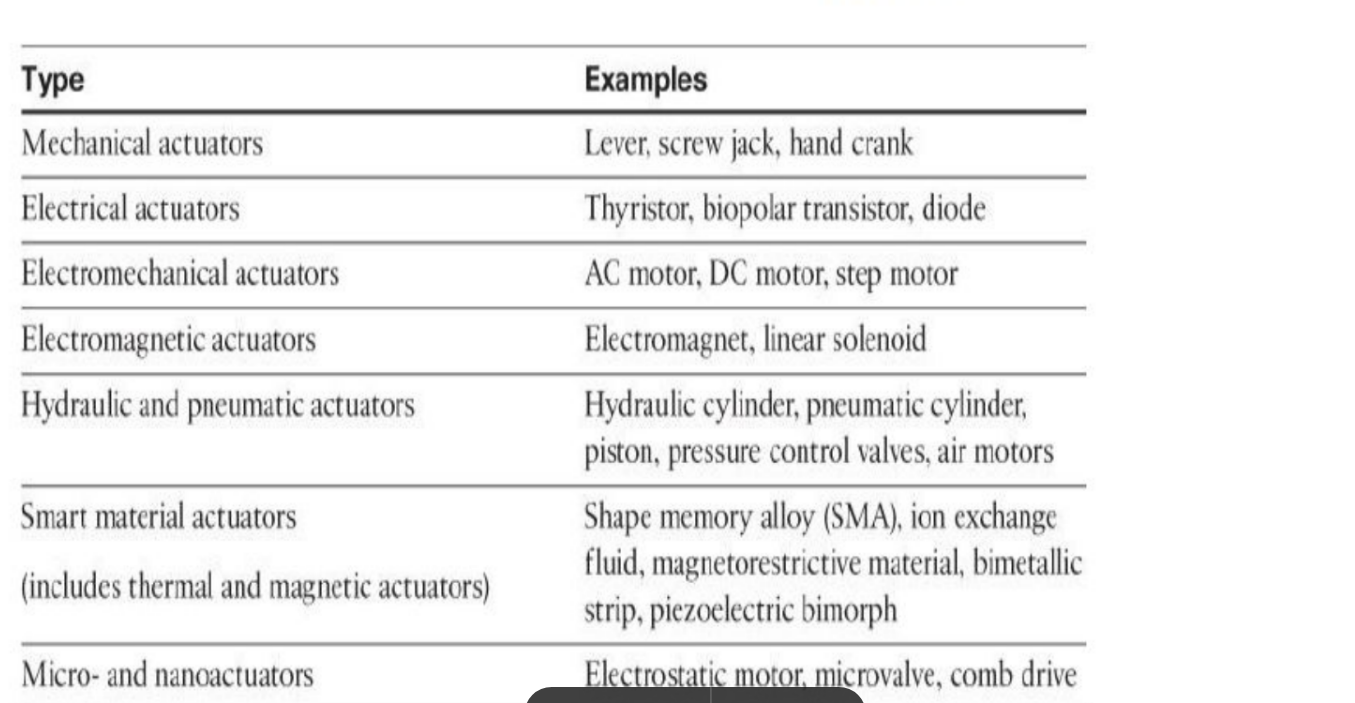


**Actuators: Analog and Digital actuators, like relay, Switch, etc.**

* Actuators are natural complements to sensors
* Sensors are designed to sense and measure practically any measurable variable in the physical world.
* They convert their measurements (typically analog) into electric signals or digital representations that can be consumed by an intelligent agent (a device or a human).
* Actuators, on the others hand, receive some type of control signal (commonly an electric signal or digital command) that triggers a physical effect, usually some type of motion, force, and so on.
* Sensors provide the information, actuators provide the action



* Actuators also vary greatly in function, size, design, and so on.
* Some common ways that they can be classified include the following:
* Type of motion: Actuators can be classified based on the type of motion they produce (for example, linear, rotary, one/two/three-axes).
* Power: Actuators can be classified based on their power output (for example, high power, low power, micro power)
* Binary or continuous: Actuators can be classified based on the number of stable-state outputs.
* Area of application: Actuators can be classified based on the specific industry or vertical where they are used.
* Type of energy: Actuators can be classified based on their energy type.



Protocols in IoT: RFID and NFC (Near-Field Communication),

What is RFID?

▪ RFID = Radio Frequency IDentification.

▪ An ADC (Automated Data Collection) technology that:

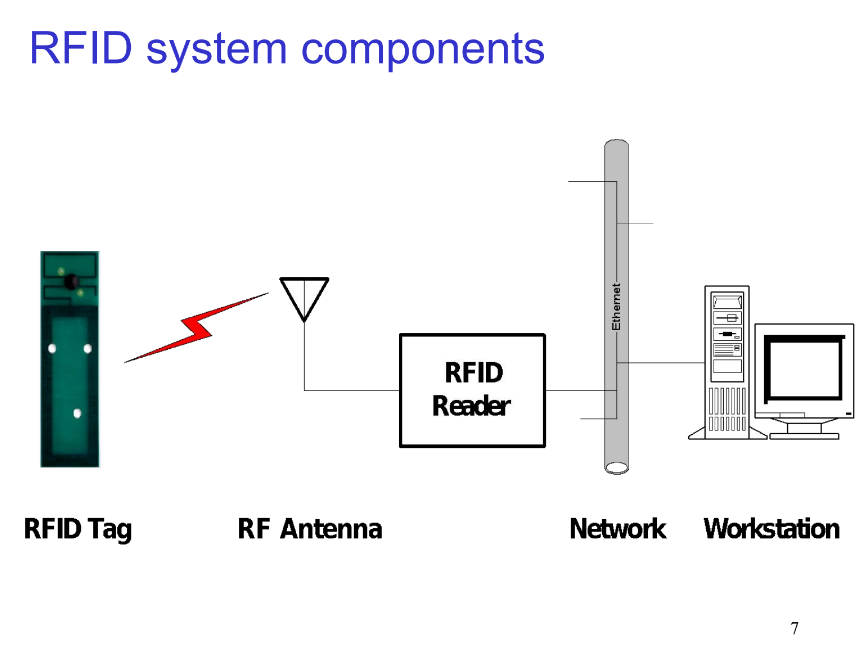
– uses radio-frequency waves to transfer data between a reader and a movable item to identify, categorize, track..

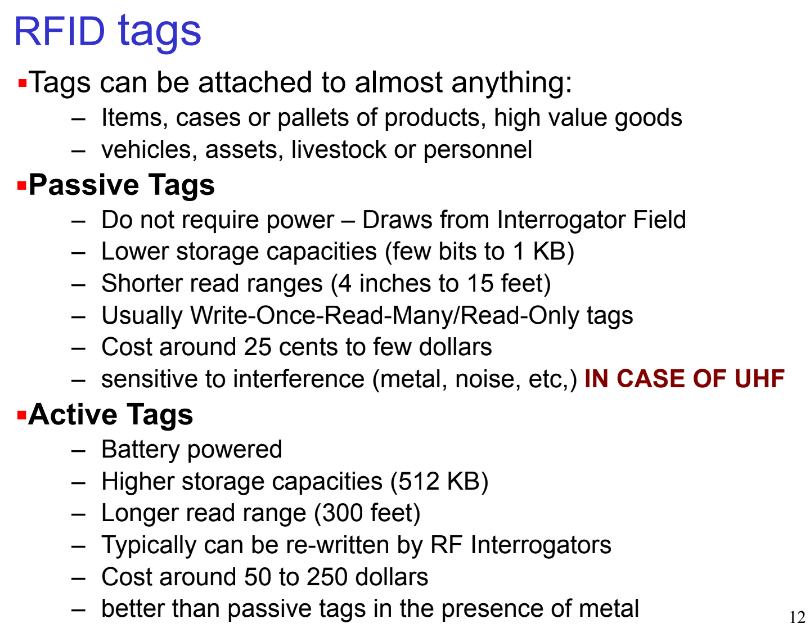
– Is fast and does not require physical sight or contact between reader/scanner and the tagged item.

– Performs the operation using low cost components.

– Attempts to provide unique identification and backend integration that allows for wide range of applications.

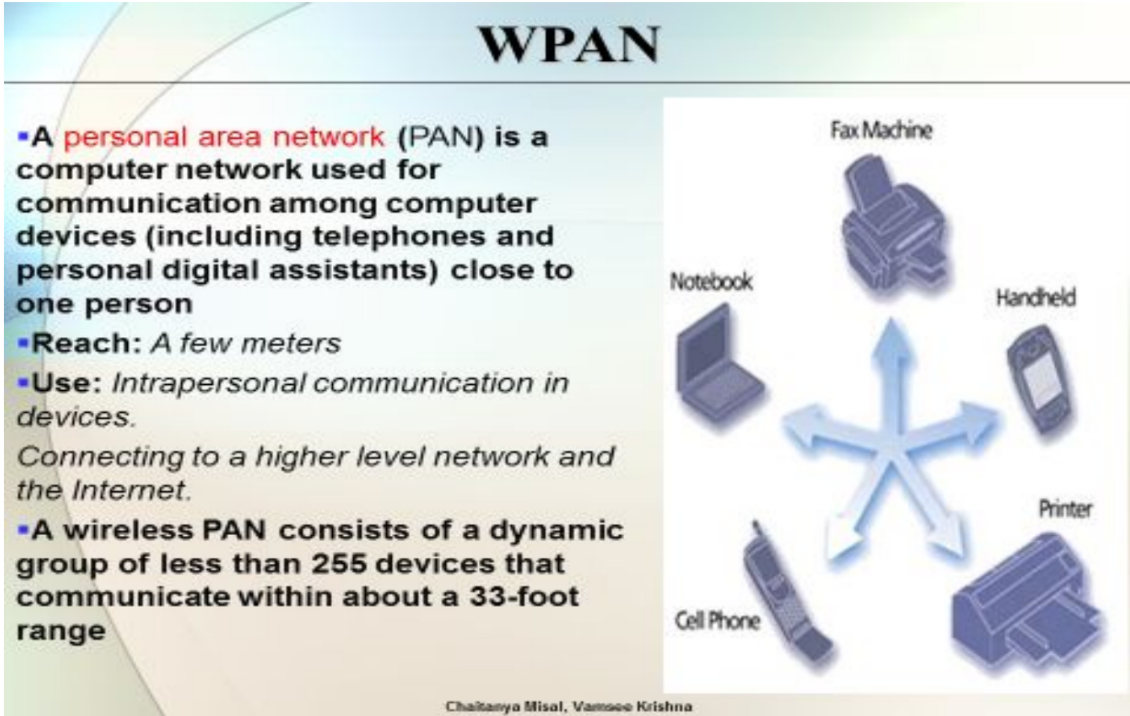
▪ Other ADC technologies: Bar codes, OCR (Optical character recognition).



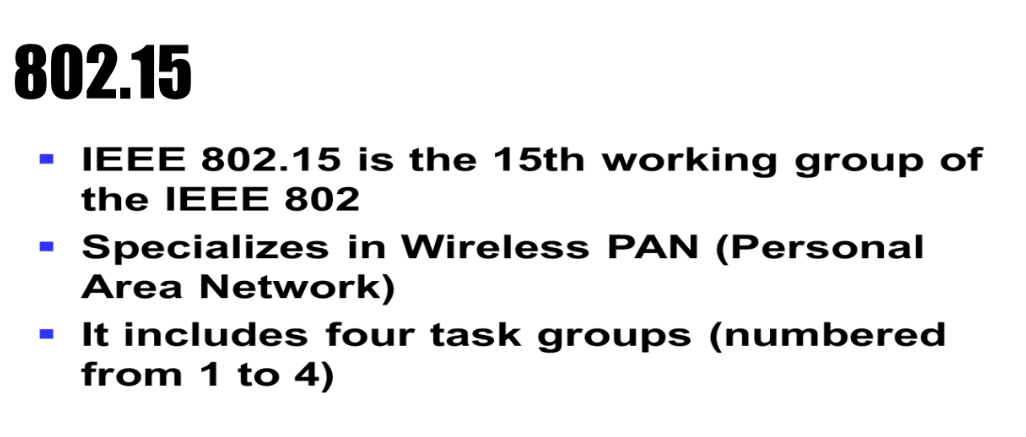


Bluetooth Low Energy (BLE) roles, LiFi,

WPAN



std: 802.15



standards: Bluetooth,

Bluetooth and BLE are radio protocols for Personal Area Network(PAN)

Mostly these are on a person's body or in close proximity to them

Typical range: very short, 20m (or less)

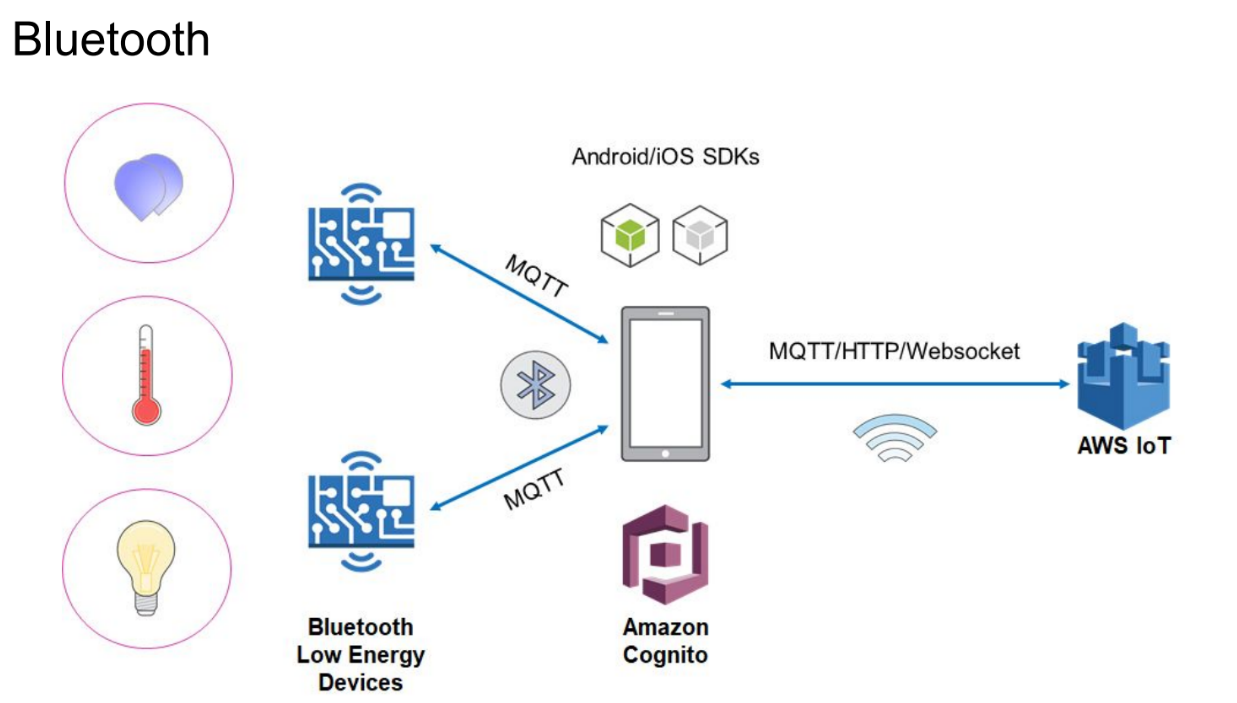
Max output power: very limited, 0.003 W

Bandwidth: limited, 0.7-2.1 Mbit/s

Security: pairing task to exchange encryption keys

Cost: cheap equipment, no transmission costs

Good for: devices that stay in close proximity of each other, like between a smartphone and a headset, heart rate monitor, bicycle speedometer



**IEEE 802.15.4.**

IEEE 802.15.4 is a low-cost, low-data-rate wireless access technology for devices that are

operated or work on batteries. This describes how low-rate wireless personal area networks

(LR-WPANs) function.

Properties:

1. Standardization and alliances: It specifies low-data-rate PHY and MAC layer requirements for wireless personal area networks (WPAN).

IEEE 802.15. Protocol Stacks include:

ZigBee:

a. ZigBee is a Personal Area Network task group with a low rate task group 4.

b. It is a technology of home networking.

c. ZigBee is a technological standard created for controlling and sensing the network.

d. ZigBee is the Personal Area network of task group 4 so it is based on IEEE 802.15.4 and

is created by Zigbee Alliance.

6LoWPAN: stands for IPv6 over Low-power Wireless Personal Area

Networks

a. The 6LoWPAN system is used for a variety of applications including wireless sensor networks.

b. This form of wireless sensor network sends data as packets and uses IPv6 - providing the

basis for the name - IPv6 over Low power Wireless Personal Area Networks.

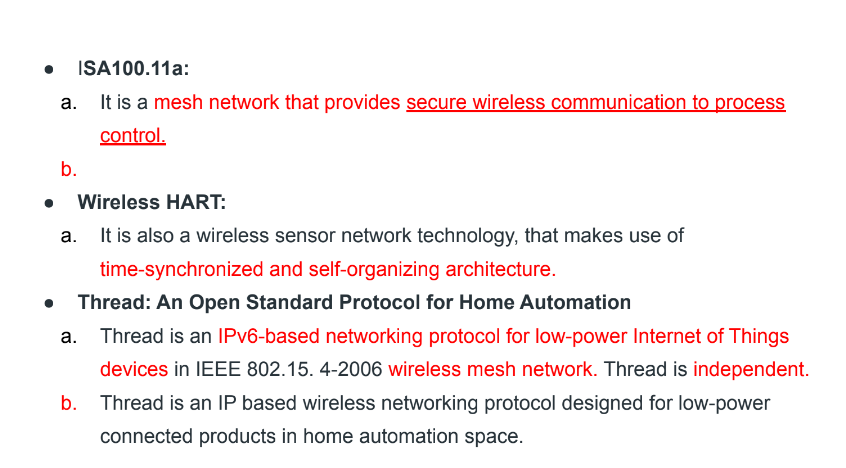
ZigBee IP:

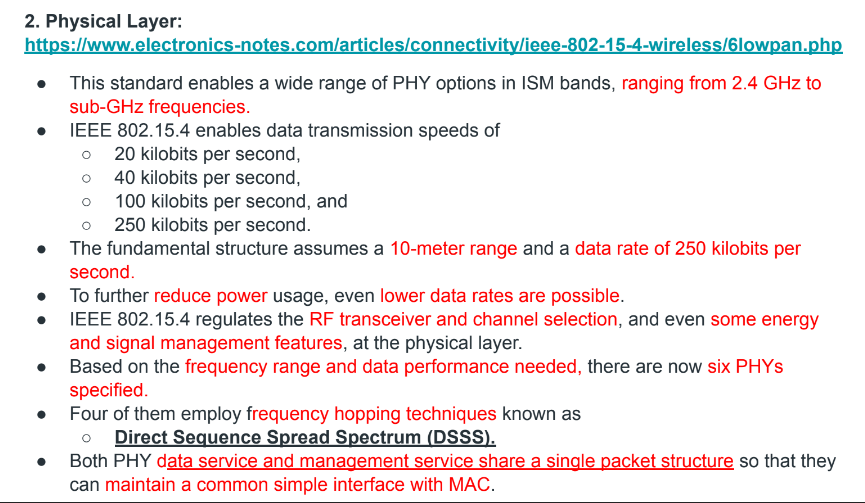
a. Zigbee is a standards-based wireless technology that was developed for low-cost

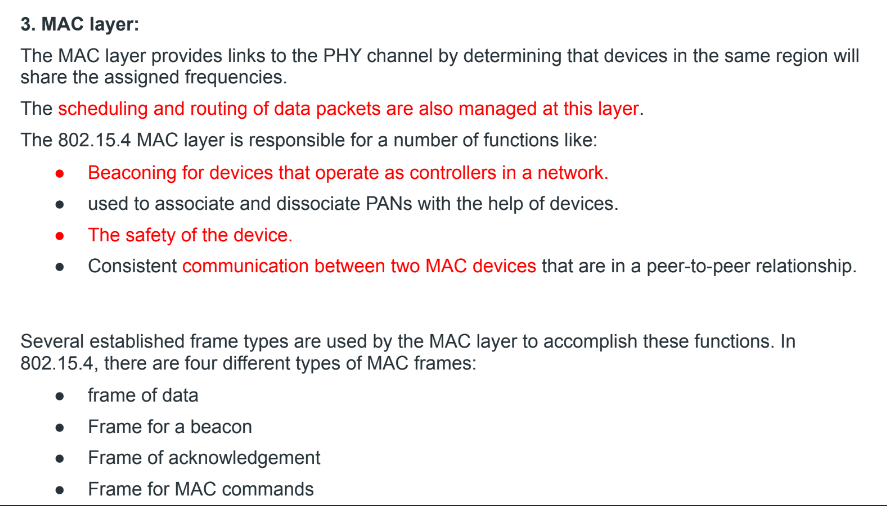
and low-power wireless

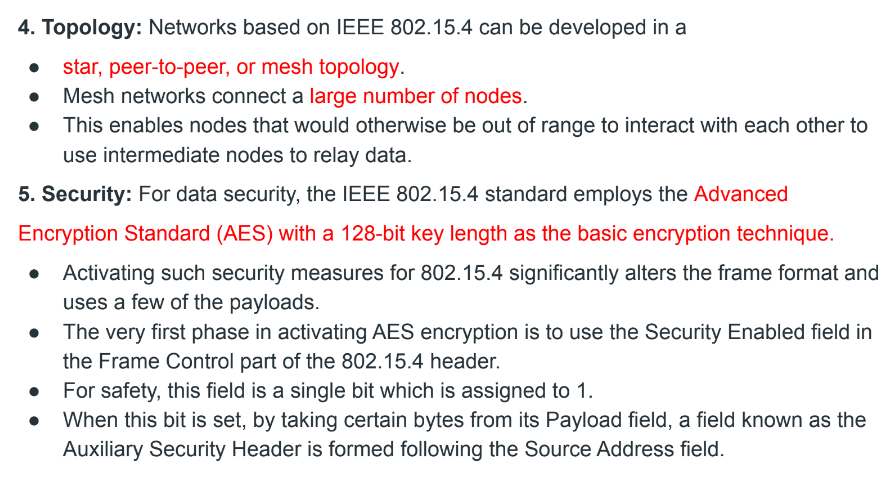
i. machine-to-machine (M2M) and

ii. internet of things (loT) networks.





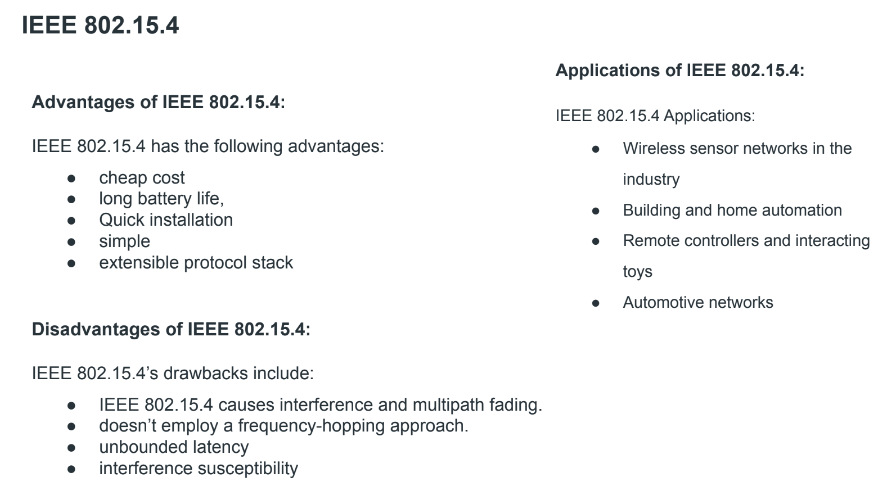


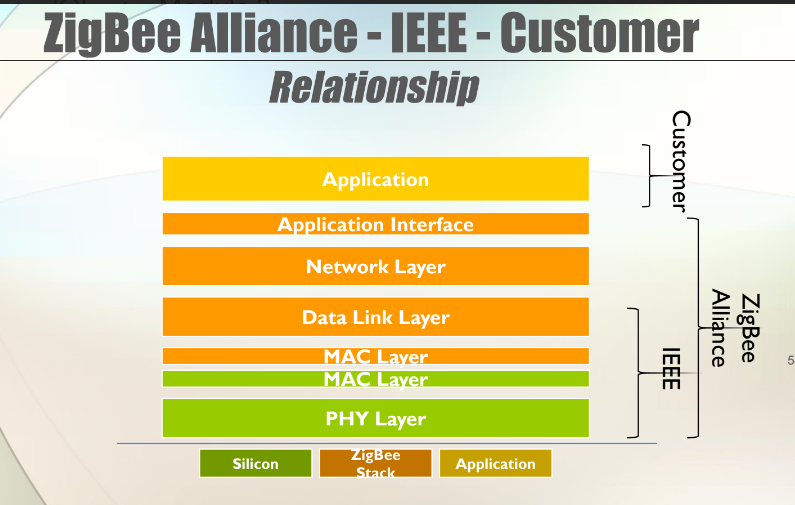


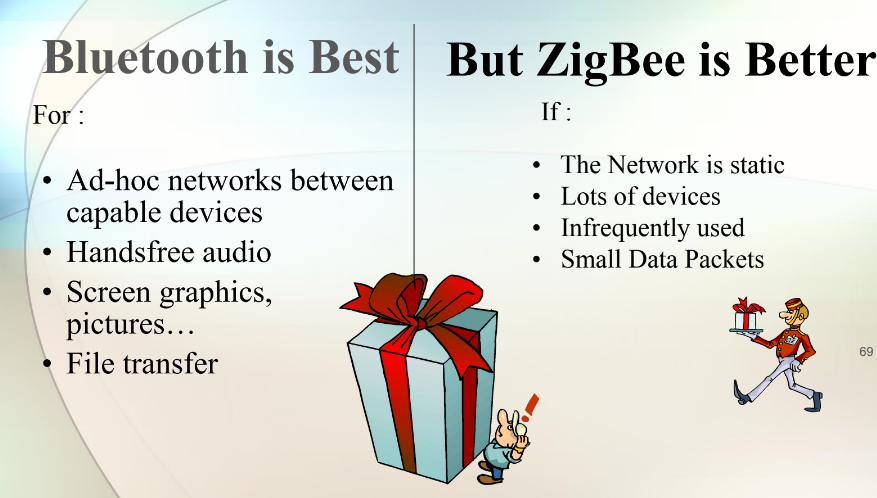
6. Competitive Technologies: The IEEE 802.15.4 PHY and MAC layers serve as a

basis for a variety of networking profiles that operate in different IoT access scenarios.

DASH7 is a competing radio technology with distinct PHY and MAC layers.

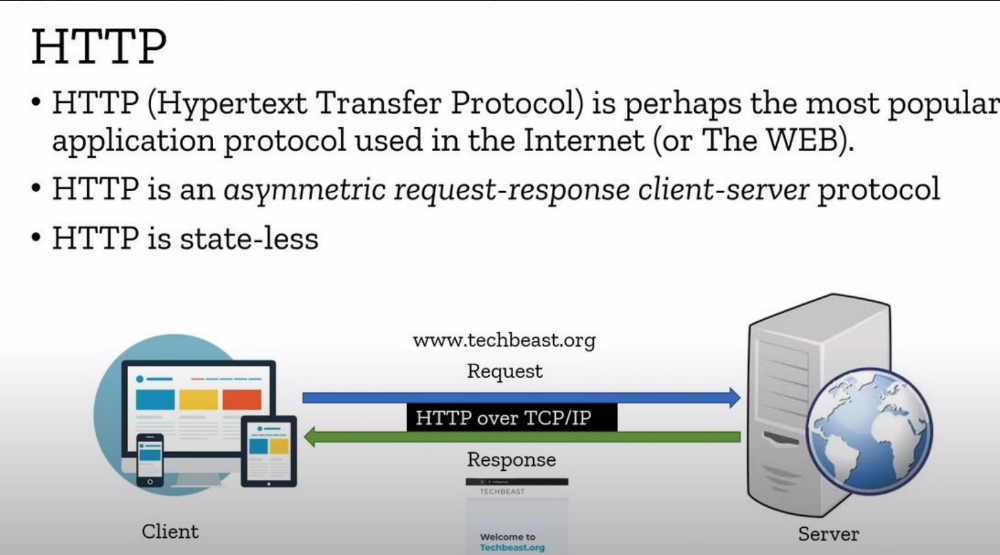


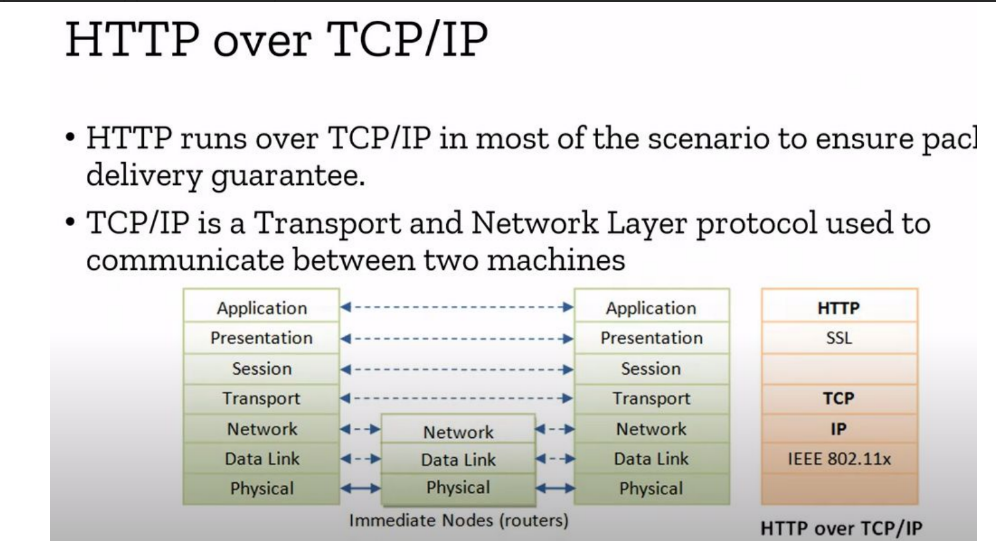


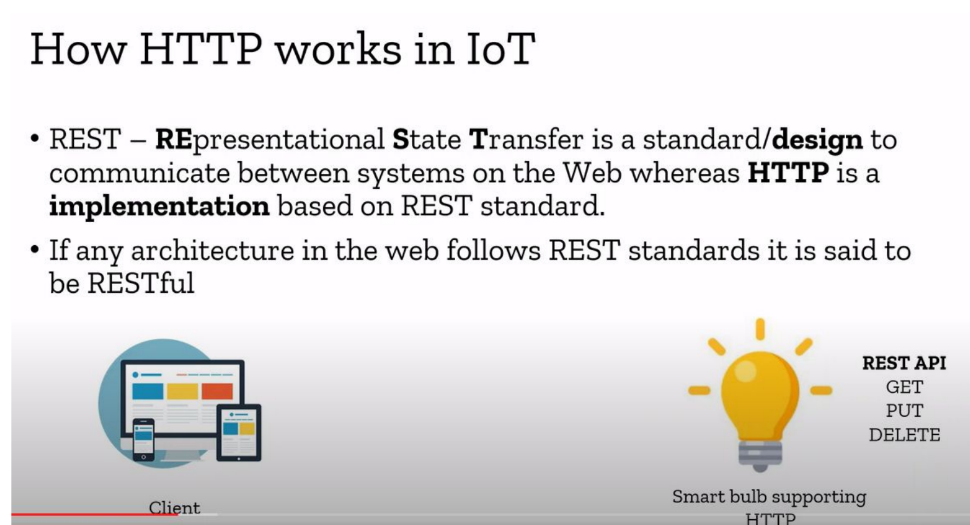


**Module 4**: Edge to Cloud Protocol

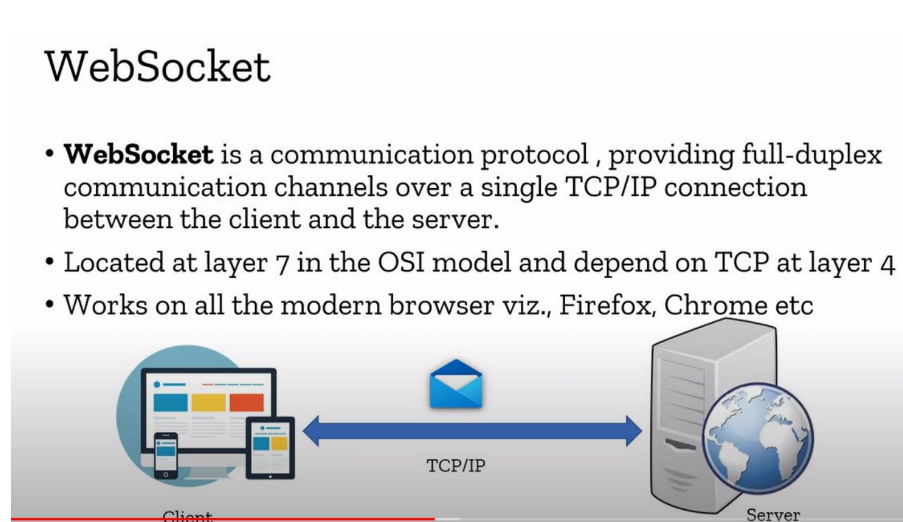
**HTTP,**

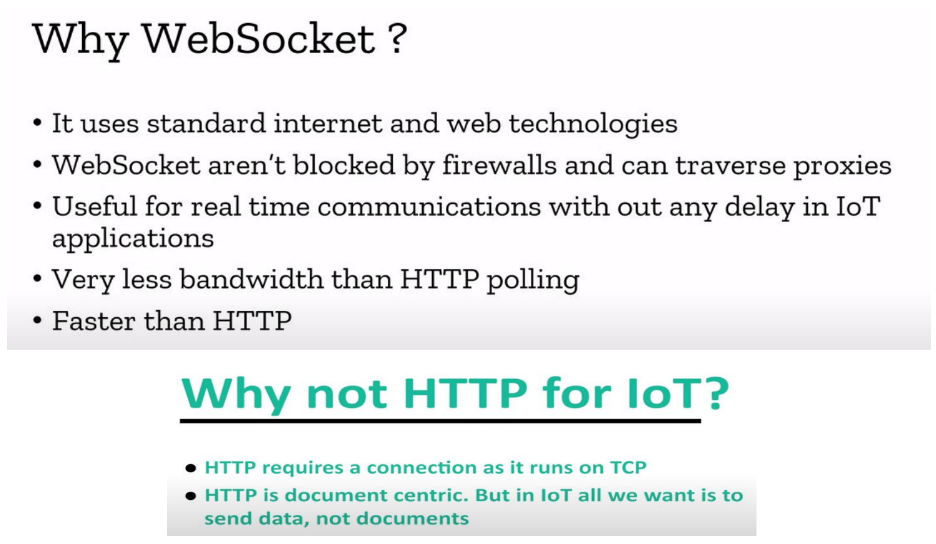






**WebSocket,**





Platforms.

HTTP - MQTT -. Complex Flows:

**IoT Patterns**: Real-time Clients,

The **real-time client pattern** involves connecting client devices that must receive data or updates instantly, without delay, often for monitoring, control, or alerting purposes.

**Message queuing telemetry transport (MQTT)**

● Perhaps the most widely used protocol in industry level IoT products, MQTT uses a

publish/subscribe model for transacting data between different devices.

● In this model, devices are divided into three types, namely, the producers, consumers and

brokers.

● Producers publish data under what are called topics.

● A topic is basically a channel under which related data can be published by the producer.

● The consumer devices subscribed to that particular channel then receive the message.

● The entire transmission of data from a publisher to its subscribers is made possible with the

help of brokers.

● Brokers receive messages from producers that they then filter into designated topics and

share with the subscribers for each of the topics.

● MQTT uses the TCP/IP protocol to enable communication between multiple devices.

● What makes it ideal for industry level products is that it simplifies sending and receiving

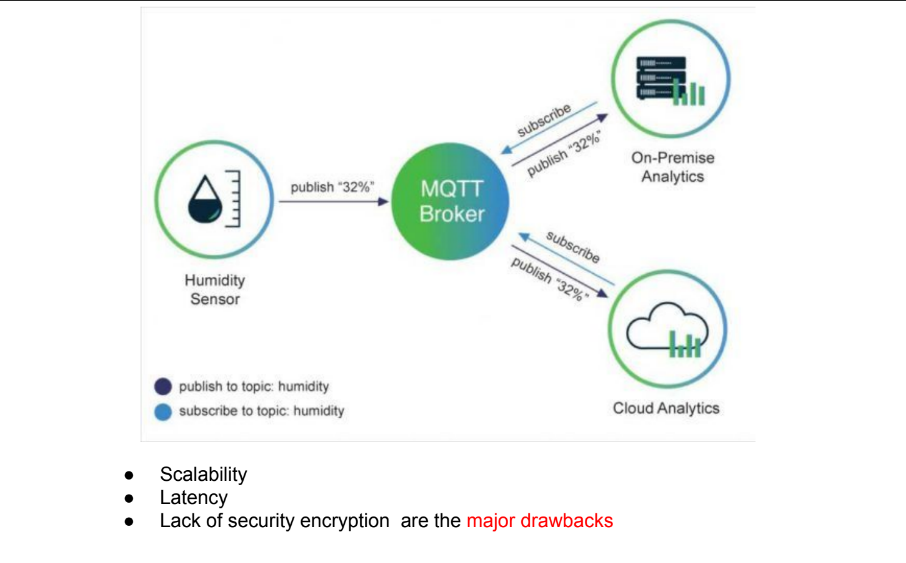
messages between multiple devices.

● In the example, the humidifier is the producer that publishes data to the broker under

separate topics. The consumers, which in this example are the on-premise analytics and the

cloud, receive the data from the broker for their respective topics. So the producer publishes

the data to the broker, and the broker publishes the data to the subscribers.



**MQTT-SN,**

* MQTT-SN (MQTT for Sensor Networks) is an extension of the widely adopted MQTT protocol, tailored for sensor networks. It addresses the unique requirements of resource-constrained devices, making it a key player in various IoT applications.
* MQTT-SN is a Publish/Subscribe message transfer protocol designed for WSN (Wireless Sensor Networks), which is aimed to provide application layer communication standards for non-TCP/IP embedded devices (such as Zigbee and Bluetooth).



There are three common architectures in MQTT-SN protocol deployment:

* + Client and Gateway are deployed in a same LAN (such as Zigbee) to communicate via MQTT-SN protocol, and Gateway reports data to the MQTT broker in the cloud through Ethernet and MQTT protocol.
  + MQTT Broker and MQTT-SN Gateway are integrated and deployed in the cloud. Client communicates directly with the MQTT-SN Gateway in the cloud through UDP and MQTT-SN.
  + The third deployment mode is similar to the first one with the difference that the MQTT-SN protocol is used for interaction with the MQTT-SN Gateway in the cloud.

**Constrained Application Protocol (CoAP)**

● When it comes to communication over the Internet, there is no protocol like the HTTP.

● With its underlying REST framework and simplicity, HTTP is highly reliable and easy to

implement.

● The trouble however is that it can prove heavy for most IoT devices, as these

generally require a more lightweight protocol.

● So can there be a protocol that has all the goodness of HTTP in addition to being lightweight and easy to implement?

● That is the problem that the IETF (Internet Engineering Task Force) Constrained RESTful environments working group tried to solve by creating CoAP in 2013.

● CoAP uses the user datagram protocol (UDP) for establishing communication between IoT

devices, unlike HTTP which uses TCP.

● CoAP uses a RESTful architecture and uses the HTTP GET, POST, PUT and DELETE methods.

● The beauty of CoAP, which distinguishes it from HTTP, is that UDP allows the IoT devices to communicate without having to establish a prior connection.

● This means that an IoT device using CoAP can relay multiple messages to multiple

devices without having to wait for a handshake.

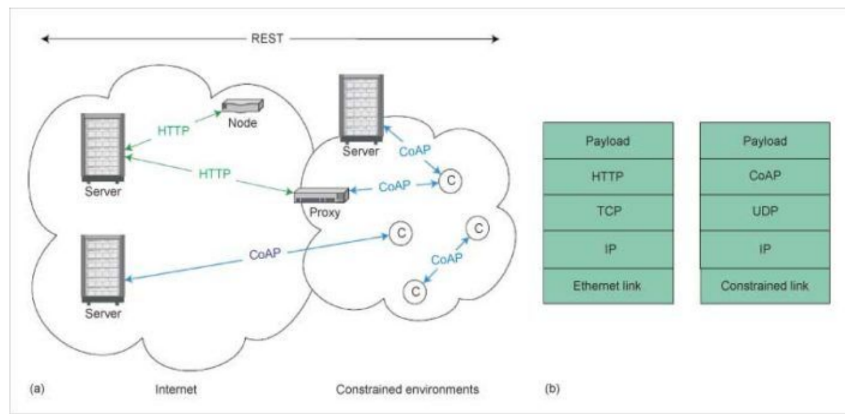
● This makes the communication faster while continuing to use very low bandwidth.

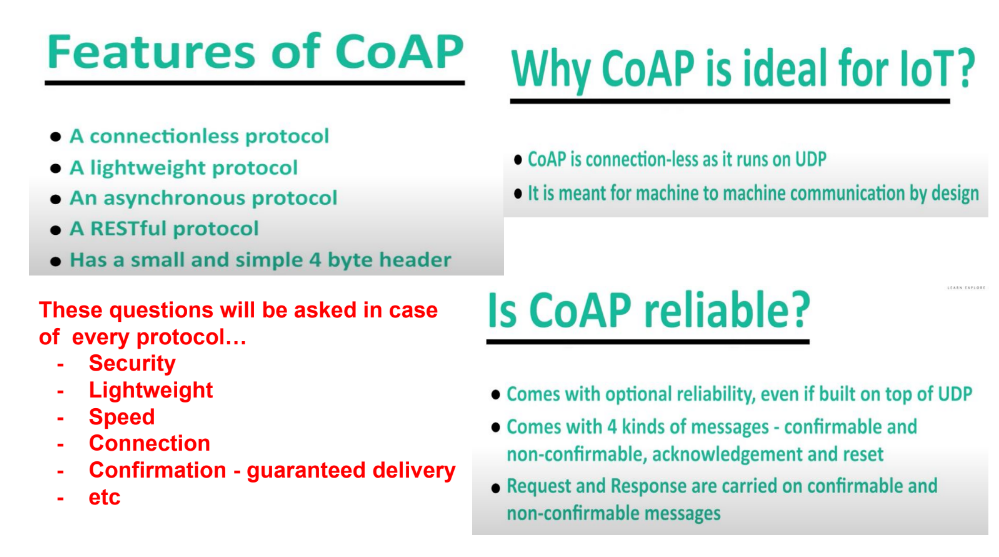
● To mitigate the risk of messages getting dropped due to a lack of connection, CoAP uses a

confirmable/non-confirmable feature, using which messages that have been delivered can be marked confirmable.

● CoAP also uses datagram transport layer security (DTLS) for secure exchange of

messages in the transport layer.





**Streaming Text Oriented Message Protocol (STOMP),**

STOMP is the Simple (or Streaming) Text Orientated Messaging Protocol, formerly known as TTMP. It provides an interoperable wire format that allows STOMP clients to talk with any message broker asynchronously supporting the protocol. It is similar to HTTP, and works over TCP using the commands – CONNECT, SEND, SUBSCRIBE, UNSUBSCRIBE, BEGIN, COMMIT, ABORT, ACK, NACK, DISCONNECT.

STOMP is a simple and easy-to-translate protocol. Many developers managed to write an STOMP client in just a couple of hours in their particular language, runtime, or platform into the STOMP network. So if the language/runtime we prefer does not offer a good enough STOMP client we can write one without any hassle.

STOMP Client Commands

In STOMP communication between client and server is through a “frame” consisting of a number of lines. The first line contains the command, followed by headers in the form <key>: <value> (one per line), followed by a blank line and then the body content, ending in a null character. Communication between server and client is through a MESSAGE, RECEIPT, or ERROR frame with a similar format of headers and body content.

**Advanced Message Queuing Protocol (AMQP),**

● AMQP is another publish/subscribe model based protocol that is primarily focused

on guaranteed completion of transactions. Designed back in 2003, it was used

mostly in the financial sector for obvious reasons.

● AMQP makes use of advanced queuing and distribution techniques to ensure

security, reliability and speed.

● the publisher publishes the message to the AMQP broker with a predesignated

routing key attached to the message.

● Utilising the routing key, the broker queues the message for delivery to the

designated consumer.

● These messages are bound to exchanges with the help of bindings.

● There are, however, a few drawbacks of the AMQP model which has restricted its mass

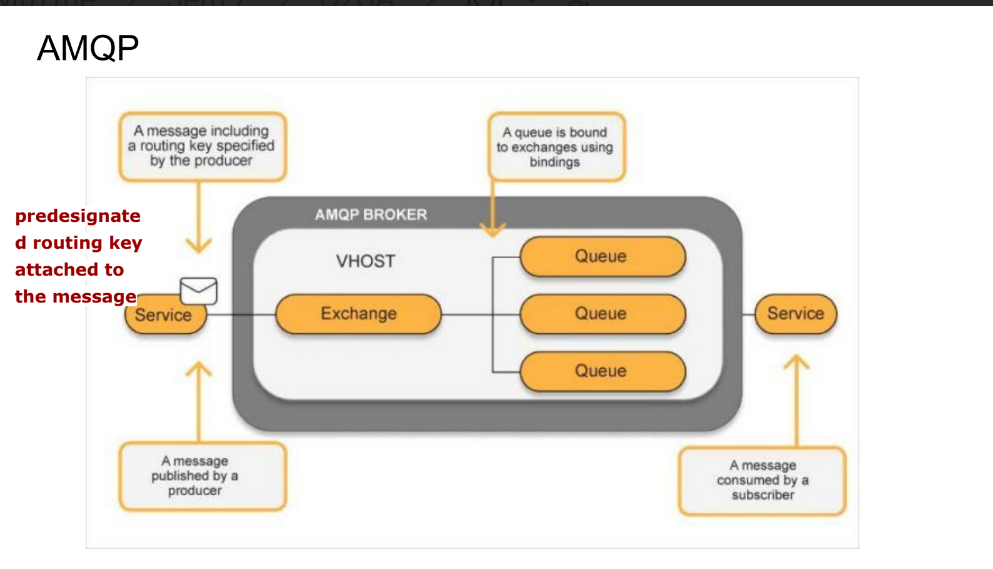
adoption in the IoT space.

○ For one, it is a heavy protocol that is not ideal for smaller IoT devices with a

lack of bandwidth and power.

○ Second, the guaranteed delivery of messages is not always a high priority

for all IoT projects.



Comparison of Protocols.

1. If using a RESTful framework with all the goodness of HTTP is a major requirement for your project, then you cannot go wrong with CoAP. CoAP provides all the features of HTTP, with the additional speed and reliability provided by UDP and message labelling.

2. If your IoT devices have major battery related constraints and you require a lightweight protocol, then MQTT is more suited for your project. It is important to keep in mind that MQTT does suffer with latency issues; so use this protocol only if high latency is not a major concern for your project.

3. If your project is data-intensive and requires fast transaction of messages between multiple devices, then XMPP (Extensible Messaging and Presence Protocol ) might be a protocol worth considering. But do this only if the security of the data is not a major concern as XMPP does not have any inbuilt security features. Needless to say, if your project uses XML, then the go-to protocol should be XMPP.

4. If what you are working on are large, high-performance sensor networks such as the ones used in self-driving cars and smart air-traffic control systems, then DDS is probably the ideal choice.

However, due diligence is necessary before selecting any protocol, and this is even more true when you are dealing with such advanced projects. DDS is a heavyweight protocol, and requires your devices to have a decent battery life and bandwidth to be able to function.

The Data Distribution Service for Real-Time Systems (DDS) is an Object Management Group (OMG) Machine-to-machine middleware "m2m" standard that aims to enable

scalable, real-time, dependable, high-performance and interoperable data exchanges

between publishers and subscribers.

5. AMQP is a protocol that can be considered for projects that are end-to-end applications — for instance, heavy industrial machinery that has the capacity to handle data-intensive tasks without the concern for power consumption. If transaction completion is an absolute necessity for your project, then AMQP is the most reliable protocol for it.

6. LwM2M is the ideal protocol for telemetry and device management based IoT projects. You can create industry-level large scale IoT projects with high scalability, speed and security. You can consider using it for a wide array of projects with varying requirements and

consumer bases.

**Which IoT protocol is right for your project?**

● What is interesting about IoT, unlike other tech projects, is that the requirements

vary a lot from project to project, not only in terms of hardware but also

software.

● For instance, an internal lighting system might require a connection with the

cloud while another similar lighting system might require a more localised

solution.

● In some projects, scalability and speed are the top priorities while for others,

guaranteed completion of transactions and security might be the focus.

● As discussed in earlier sections, protocols play a critical role in facilitating such

requirements.

● If the hardware is the body, the protocol is the brain of the device.

● Keeping this in mind, it is recommended to do a detailed audit and review of

the project requirements.

● It is also important to research industry level IoT projects similar to yours and see

which protocol they use. Here’s a list of points to consider in this regard.

**Module 5**: IoT and Data Analytics

**Defining IoT Analytics,**

IoT Analytics challenges,

IoT analytics for the cloud,

Strategies to organize Data for IoT Analytics,

Linked Analytics Data Sets,

Managing Data lakes,

The data retention strategy,

visualization and Dashboarding.

**Module 6:** IoT Application Design

Prototyping for IoT and M2M

Case study related to: Home Automation (Smart lighting,

home intrusion detection), Cities (Smart Parking), Environment

(Weather monitoring, weather reporting Bot, Air pollution

monitoring, Forest fire detection, Agriculture (Smart

irrigation), Smart Library.

Introduction to I-IoT, Use cases of the I-IoT, IoT and I-IoT –

similarities and differences,

Introduction to Internet of Behaviour (IoB).