(NILESH SIR's PPT):

History of IoT : The term Internet of Things is 23 years old. First time the term "Internet of Thing" is coined by "Kevin Ashton"(1999).

What is a thing (in the internet of things)? A thing, in the context of the internet of things (IoT), refers to any entity such as a device that forms a network and can transfer data with other devices over the network.

What is the Internet? In simple words, it's a Network of Networks or Interconnected LANs (OR) ,The Internet is a vast network that connects computers all over the world.

Define Term IOT? (https://youtu.be/APH6Nrar27w)

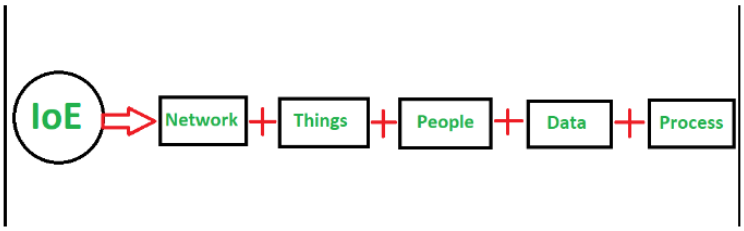
The Internet of things (IOT) describes the network of physical objects - "things" - that are embedded with sensors, software and other technologies for the purpose of connecting and exchanging data with other devices and systems over the internet.

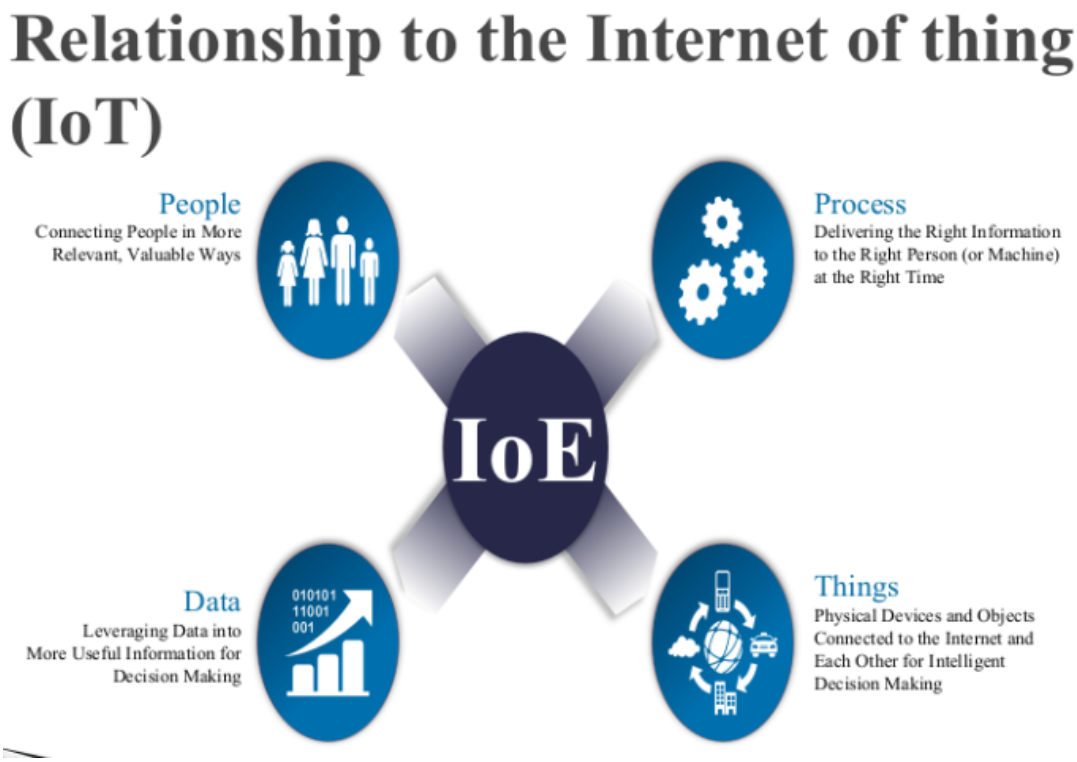
IoT is a network of interconnected computing devices which are embedded in everyday objects, enabling them to send and receive data.

The IoT is not just limited to the connected or networked devices, but in a broad way IoT devices exchange meaningful information from one device to another to get desired results.

Internet of Things has two main parts i.e **'Internet'** which is the backbone of connectivity and **'Things'** meaning object/physical devices.

Define IOE? IoE(Internet of EveryThing) is the intelligent connection between 4 key elements i.e people, process, data, and things. It is considered as superset of Internet of Things (IOT).





Difference between IoE and IoT :

IoE	IoT
IoE is the intelligent connection between people, process, data and things by creating 'web of things' which is the next generation of internet.	IoT is the network of physical devices where collection and exchange of data occurs without human intervention.
In IoE, communication occurs between Machine to Machine, Machine to People and technology assisted People to People.	In IoT, communication occurs between Machine to Machine.
IOE focuses on connecting everything, including people, processes, data, and things.	IoT focuses on connecting physical devices, sensors, and other electronic equipment.
Includes components like humans, processes, data, and devices	It Primarily involves devices and things
More complex due to involving various components	Relatively simpler as it deals with devices
Highly interdependent components	Components can be somewhat independent
IOE requires more robust security measures to protect the exchange of sensitive data between people, processes, and things.	IoT also requires security measures, but the focus is on securing devices and the data they generate.
IOE is not yet a standard term or concept, and there are no established standards.	IoT has established standards, protocols, and frameworks that govern device connectivity, data exchange, and security.
Examples Smart cities, healthcare systems, connected industries	Smart thermostats, wearable devices, smart appliances
It is considered as the superset for Internet of Things(IoT) and it is considered a generation after IoT.	It is considered as the subset of bigger Internet of Everything(IoE) and IoT is considered one generation before IoE.

IOT vs WOT: https://youtu.be/eKB1mSvuiQA?si=8zAS_26uri8vD9LC

What is ubiquitous computing? (OR) What is Pervasive Computing? : Pervasive Computing is also called as Ubiquitous computing, and it is the new trend toward embedding everyday objects with microprocessors so that they can communicate information. It refers to the presence of computers in common objects found all around us so that people are unaware of their presence. All these devices communicate with each other over wireless networks without the interaction of the user.
The words pervasive and ubiquitous mean **"existing everywhere"**

Major Components of IoT : There are 5 major components of IoT (Internet of Things)

1. Sensors or Devices: Sensors or Devices are basically used to collect and transmit the data and also perform actions based on those data.For example, the sensors can be used for measuring temperature and humidity.
2. Gateway: Gateway is also a device component that basically acts as an intermediate between the sensors and the central cloud. Gateway is one of the essential components of IoT that offers communication, management, and data processing.
3. Cloud: Cloud in IoT refers to the service that provides the management, storage, and processing of the data that is generated by IoT (Internet of Things) devices.
4. Analytics: The data after being received in the cloud processing is done . Various algorithms are applied here for proper analysis of data (techniques like Machine Learning etc are even applied). In analytics, meaningful insights are analyzed that are generated by IoT devices and sensors.

5. User Interface: User Interface, also known as UI in the Internet of Things (IoT) and provides an interface by which the users can interact with the applications and systems.

Communication Technology for IoT:

Bluetooth: Bluetooth is a short-range wireless communication standard used for connecting devices in close proximity. It's suitable for applications like wireless headphones, smartwatches, and connecting peripherals to smartphones.

Range: Short to Medium

Data Rate: Moderate

Use Cases: Wearables, smart homes, healthcare devices.

Advantages: Low power, low cost, widely supported in consumer electronics.

Zigbee:

Zigbee is designed for low-power, low-data-rate applications. It operates on the 2.4 GHz frequency and is commonly used in smart home devices, industrial automation, and sensor networks.

Range: Short to Medium

Data Rate: Low to Moderate

Use Cases: Home automation, industrial automation, smart energy.

Advantages: Low power, low cost, mesh networking for extended range.

Z-Wave:

Z-Wave is a wireless communication protocol specifically designed for home automation. It operates on sub-1 GHz frequencies, providing good range and penetration through walls.

Range: Short to Medium

Data Rate: Low to Moderate

Use Cases: Home automation, smart security systems.

Advantages: Low power, dedicated frequency band for IoT applications.

NFC (Near-Field Communication):

NFC is used for close-range communication between devices. It's commonly found in smartphones and is used for applications like mobile payments, smart cards, and pairing devices.

Range: Very Short (up to 10 cm)

Data Rate: Low to Moderate

Use Cases: Contactless payments, access control, data transfer between devices.

Advantages: Simple, secure, and widely used for short-range applications.

WiFi:

WiFi (Wireless Fidelity) provides high-speed wireless connectivity over a local area network. It's widely used for connecting IoT devices in homes, offices, and industrial environments.

Range: Medium to High

Data Rate: High

Use Cases: Home and office networking, smart cities, industrial applications.

Advantages: High data rates, wide adoption, supports IP-based communication.

RFID (Radio-Frequency Identification):

RFID uses radio-frequency signals to identify and track objects or individuals. It's commonly used in logistics and supply chain management for tracking goods.

Range: Short to Medium

Data Rate: Low

Use Cases: Supply chain management, asset tracking.

Advantages: Simple and cost-effective for identification and tracking.

2G/3G/LTE:

Cellular networks (2G, 3G, LTE) provide wide-area wireless connectivity. They are suitable for applications that require reliable and high-speed communication over long distances.

Range: Wide (depends on the generation)

Data Rate: Moderate to High

Use Cases: Industrial IoT, smart cities, connected vehicles.

Advantages: Wide coverage, high data rates, seamless mobility.

Wibro/Mobile WiMax:
WiBro (Wireless Broadband) and Mobile WiMax are wireless broadband access technologies that offer high-speed internet access over a wide area. They are used in applications where high data rates are essential.
Range: Medium to High
Data Rate: High
Use Cases: Broadband wireless access, smart cities.
Advantages: High data rates, long-range coverage.
PLC (Power Line Communication):
PLC enables data transmission over existing power lines, making it convenient for applications where dedicated communication infrastructure is challenging.
Range: Medium
Data Rate: Moderate
Use Cases: Smart grid, home automation.
Advantages: Utilizes existing power infrastructure for communication.
Ethernet:
Ethernet is a widely used wired communication technology, offering high data rates and reliability. It is commonly employed in industrial environments and for backhaul connections in IoT deployments.
Range: Medium to High
Data Rate: High
Use Cases: Industrial automation, smart buildings, wired networks.
Advantages: Reliable, high-speed communication over wired networks.

Issues and challenges of IoT :

- Security and Privacy Concerns: IoT devices can be vulnerable to cyberattacks and hacking due to insufficient security measures.
- Power Consumption: Many IoT devices are battery-powered, and optimizing power consumption is crucial for extending device lifetimes and reducing maintenance requirements.
- Scalability: As the number of IoT devices grows exponentially, managing and scaling the infrastructure required to support these devices becomes increasingly complex.
- Data Management and Analytics: Handling and processing the massive volumes of data generated by IoT devices can be overwhelming.
- Cost of Implementation: Developing, deploying, and maintaining IoT systems can be expensive, especially for small and medium-sized businesses or organizations with limited resources.
- Lack of Skill and Expertise: The rapid growth of IoT has resulted in a shortage of skilled professionals who understand both the hardware and software aspects of IoT systems.
- Standards: The lack of unified standards in the IoT ecosystem leads to interoperability challenges and hinders seamless communication among diverse devices and platforms.

ZigBee: https://youtu.be/_dl8oTiL6Wg https://youtu.be/k3_bfLs7VMw
Zigbee is a wireless communication technology that is commonly used in the context of the Internet of Things (IoT). It's designed for low-power, short-range communication between devices, making it well-suited for creating networks of small, battery-powered devices that need to communicate with each other.
Basically ZigBee is a standard that is used by IoT Devices to communicate with each other.
It is based on IEEE 802.15.4 protocol and Created by the ZigBee Alliance
Operates in Personal Area Networks (PAN's) and device-to-device networks
Designed for wireless controls and sensors
Types of ZigBee Devices:

- Zigbee Coordinator Device: It communicates with routers. This device is used for connecting the devices.
- Zigbee Router: It is used for passing the data between devices.
- Zigbee End Device: It is the device that is going to be controlled.

General Characteristics of Zigbee Standard:

- Low Power Consumption: Zigbee devices are designed to operate on very low power, making them suitable for battery-operated devices.

- Short Range: Zigbee is meant for short-range communication, typically within a range of 10 to 100 meters.
- 3 frequency bands with 27 channels: Zigbee operates in multiple frequency bands, including 2.4 GHz (the most common), 900 MHz, and 868 MHz, depending on regional regulations and specific use cases.
- Low Data Rate: Zigbee is well-suited for applications that require intermittent(periodic) data transmission, such as home automation, smart lighting, and sensor networks. It's not optimized for high-bandwidth applications.
- Support Small and Large Networks (up to 65000 devices (Theory); 240 devices (Practically))

Zigbee Network Topologies:

- Star Topology (ZigBee Smart Energy): Consists of a coordinator and several end devices, end devices communicate only with the coordinator.
- Mesh Topology (Self Healing Process): Mesh topology consists of one coordinator, several routers, and end devices.
- Tree Topology: In this topology, the network consists of a central node which is a coordinator, several routers, and end devices. the function of the router is to extend the network coverage.

Advantages of Zigbee:

- Designed for low power consumption
- Provides network security and application support services operating on the top of IEEE.
- Low cost: Zigbee chips and modules are relatively inexpensive
- Mesh networking: Zigbee uses a mesh network topology, which allows for devices to communicate with each other without the need for a central hub or router. This makes it ideal for use in smart home applications
- Reliability: Zigbee protocol is designed to be highly reliable,
- Easy implementation

Disadvantages of Zigbee :

Limited range: Zigbee has a relatively short range compared to other wireless communications protocols
Limited data rate: Zigbee is designed for low-data-rate applications
Security: Zigbee's security features are not as robust as other IoT protocols

ZigBee Frequencies:

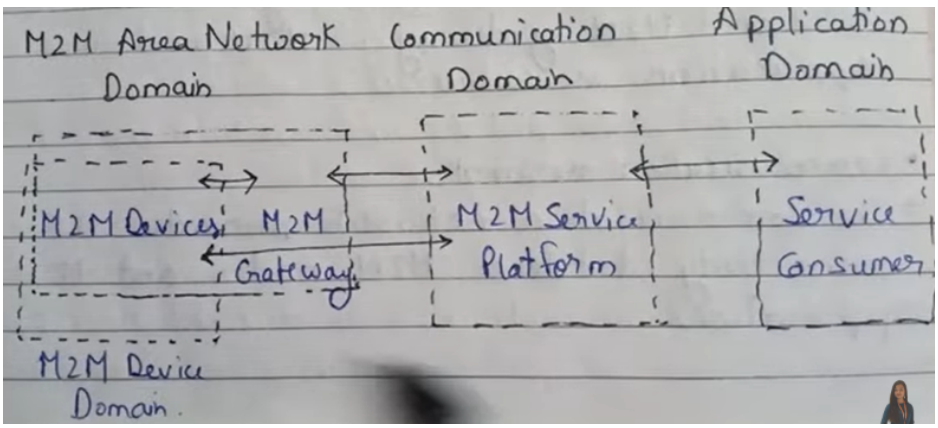
- Operates in Unlicensed Bands
- ISM 2.4 GHz Global Band at 250kbps
- 868 MHz European Band at 20kbps
- 915 MHz North American Band at 40kbps

M2M(Machine to Machine) https://youtu.be/ZW8425Tas7w
https://youtu.be/dhZUxMo4eMY (with architecture)

M2M is more recently referred to technologies that enable communication between machines without or with minimal human intervention. M2M is a subset of IoT.
- Communication in M2M may be wired or wireless .
- Point to point Communication between physical object.
- Does not follow Internet Protocol (IP)–based networks and Internet standards.

Eg: Controlling electrical devices like fans & bulbs using bluetooth from the smartphone

Architecture:



It has 4 components:
1) M2M area networks
2) Communication networks
3) Application domains
4) M2M gateways

Application of M2M:

- Security and Surveillance: M2M-enabled surveillance cameras and sensors can transmit data to cloud-based security systems. These systems can process the data to detect anomalies, trigger alerts.
- Healthcare Monitoring: Wearable medical devices can continuously monitor patients' health parameters and transmit the data to the cloud.
- Traffic Control: Traffic Control is another common area where the use of M2M communication can be seen. A traffic system collects data related to the speed and volume of the traffic with the help of various sensors and sends this information across the computers that control the devices such as signals and lights. The cameras installed on the traffic signals also collect data about the vehicles not following the traffic rules and send pictures to the software which then sends challan receipts to the defaulters.
- Banking: Banking is another common area to make use of M2M. With an increase in the smartphone market, people have started making mobile payments for their purchases.
- Tracking & Tracing :Order Management, Asset Tracking, Navigation, Traffic information.

Apart from this it is also used in Facility Management(Home / building / campus automation), Utility Companies, Smart Cities, Industrial Automation and Manufacturing etc.

Features of M2M: The ability to continually send and receive small amounts of data.

- Reduced Human Error: By automating communication between devices, M2M minimizes the potential for human error, leading to more reliable and accurate operations.
- Low Mobility: The "Low Mobility" feature of Machine-to-Machine (M2M) communication in IoT refers to the ability of M2M devices to operate effectively in scenarios where there is minimal movement or change in location
- Time-controlled: Time-controlled features in Machine-to-Machine (M2M) communication within the Internet of Things (IoT) enable devices to perform actions or exchange data based on specific time schedules or intervals.
- Low Power Consumption: The "Low Power Consumption" feature of Machine-to-Machine (M2M) communication in IoT refers to devices using minimal energy to communicate, making them suitable for battery-powered and energy-efficient applications.
- Packet switching: Packet switching in Machine-to-Machine (M2M) communication within the Internet of Things (IoT) involves breaking data into small packets before transmission. These packets are sent separately and reassembled at the receiving end. This approach is efficient, as it optimizes data transmission, reduces network congestion

Requirements for M2M:

Scalability - The M2M system should be able to continue to function efficiently as more connected objects are added.
Communication failure notification.
Continuous connectivity.
Message communication path selection - Optimization of the message communication paths within an M2M system must be possible and based on policies like transmission failures, delays when other paths exist and network costs.

IoT vs M2M:

M2M	IOT
M2M means Machine to machine communication and complete hardware based	In IoT there can be Machine to Machine communication, Machine to Sensors or Humans to machines communication and software based.
It is a point to point communication and uses non IP protocols.	It uses IP networks & protocols as the communication is multipoint
These devices don’t rely on the Internet.	Devices required internet connections.
Limited integration option devices must have corresponding communication standards.	Ultimate integration option, but requires a solution that can manage all the communication
Data can be stored locally.	Data can be stored locally and also in cloud.
Unidirectional Communication	Bidirectional Communication

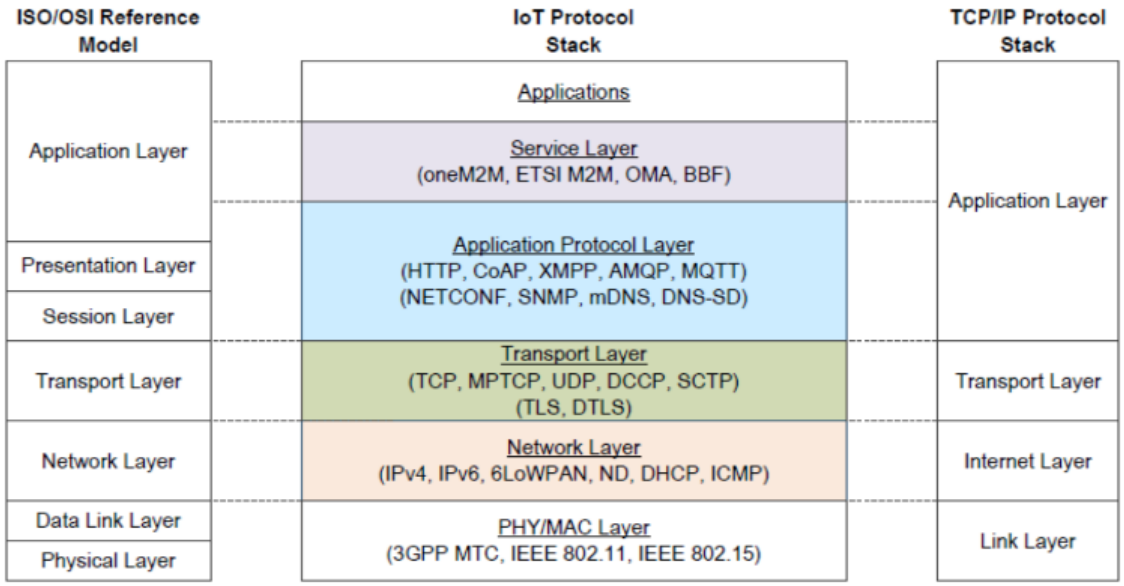
Similarities between IOT & M2M:
Both Provide remote access to machine data and both exchange info among machines without human intervention.
RFID(Radio Frequency IDentification): https://youtu.be/qUTssrhayNY
https://www.geeksforgeeks.org/introduction-of-radio-frequency-identification-rfid/
An AIDC (Automated Identification and 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.
RFID Classification:
Passive Tags: Do not require power – Draws from Interrogator Field
- Lower storage capacities (few bits to 1 KB)
- Shorter read ranges (4 inches to 15 feet)
- Usually Write-Once-Read-Many/Read-Only tags
- Cost around 25 cents to few dollars
Active Tags: Battery powered
- Higher storage capacities (512 KB)
- Longer read range (300 feet)
- Typically can be re-written by RF Interrogators
- Cost around 50 to 250 dollars

Active RFID vs Passive RFID:

Aspect	Active RFID Tags	Passive RFID Tags
Power Source	Have their own power source (battery)	Don't have their own power source
Communication Range	Longer range, often up to hundreds of meters	Shorter range, usually up to a few meters
Cost	Generally more expensive due to battery	Typically less expensive
Size	Larger in size due to the battery	Smaller in size
Read Range	Can be read from a greater distance	Read range is limited
Continuous Operation	Can operate continuously	Require an external reader to activate
Communication Speed	Faster communication due to active power	Slower communication
Lifespan	Limited by battery life	Can last for many years
Applications	Suitable for tracking high-value assets, vehicles, people	Used for inventory management, access control, etc.
Maintenance	Need periodic battery replacement	Low maintenance due to no battery
Environmental Considerations	Battery disposal and replacement	Generally more environmentally friendly

RFID advantages over bar-codes
1. No line of sight required for reading
2. Multiple items can be read with a single scan
3. Each tag can carry a lot of data (read/write)
4. Individual items identified and not just the category
5. Passive tags have a virtually unlimited lifetime
6. Active tags can be read from great distances
7. Can be combined with barcode technology

IOT Protocol and Standards:



IoT Communication Models:

Device-to-Device (D2D): D2D communication involves direct communication between two IoT devices without the need for a centralized server or cloud. Devices exchange information directly with each other.

Device-to-Server (D2S) : D2S communication involves IoT devices sending data to a centralized server or cloud platform. Devices collect data and then transmit it to a server for storage, analysis, and further processing.

Server-to-Server (S2S): S2S communication refers to communication that occurs between different servers or cloud platforms. This enables data sharing and coordination between various backend systems.

For example, in a smart home scenario:

D2D communication might involve smart devices directly exchanging information, like a thermostat communicating with a smart light to adjust brightness based on temperature.

D2S communication could be the smart devices sending data to a cloud server for remote monitoring or analytics.

S2S communication might occur between different cloud services handling various aspects of smart home automation, such as one server managing security devices and another managing energy-related devices.

IOT PROTOCOLS:

6LoWPAN: https://www.geeksforgeeks.org/what-is-6lowpan/

6LoWPAN is an IPv6 protocol, and It's extended from is IPv6 over Low Power Personal Area Network. As the name itself explains the meaning of this protocol is that this protocol works on Wireless Personal Area Network i.e., WPAN.

– 6LoWPAN is the secret sauce that allows larger IPv6 packets to flow over 802.15.4 links that support much smaller packet sizes.

Here's how 6LoWPAN is used in IoT:

IPv6 Compatibility: 6LoWPAN brings the power of IPv6 to small, low-power IoT devices. IPv6 provides a large address space, which is important as more and more devices connect to the internet.

Wireless Personal Area Networks (WPANs): 6LoWPAN is optimized for short-range wireless networks like IEEE 802.15.4, which is commonly used in home and industrial automation.

Header Compression: IPv6 headers can be relatively large, which is not ideal for small data packets from constrained devices. 6LoWPAN uses header compression techniques to reduce this overhead.

Fragmentation and Reassembly:6LoWPAN supports the fragmentation and reassembly of large packets to accommodate networks with limited frame sizes. This is important for dealing with small Maximum Transfer Units (MTUs) common in low-power wireless networks.

Energy Efficiency:The protocol is designed with energy efficiency in mind, minimizing the energy consumption of devices, which is crucial for battery-operated IoT devices.

Use Cases and Applications:
Smart Homes: 6LoWPAN is suitable for connecting and managing smart home devices such as sensors, actuators, and smart appliances.
Industrial IoT (IIoT): In industrial settings, where low-power devices are common, 6LoWPAN can be used for monitoring and control applications.
Healthcare:Wireless health monitoring devices, wearable sensors, and other healthcare-related IoT applications benefit from 6LoWPAN's low-power and efficient communication.

UDP/DTLS(Datagram Transport Layer Security)
UDP: Most IoT scenarios are well suited for UDP. UDP is a simple, connectionless transport layer protocol that operates on top of IP (Internet Protocol). It provides a way for devices to send data packets to each other without establishing a formal connection.
UDP is a much lighter protocol compared to TCP.
UDP is much faster than TCP, the header size is much smaller than TCP
UDP does not include mechanisms for ensuring that data is received correctly or in the correct order.
Since there's no need to establish a connection or maintain state, UDP can transmit data quickly.
UDP is often used for applications where speed is crucial, such as streaming media or online gaming.
Characteristics:
Low Overhead: UDP has less overhead than TCP, making it suitable for scenarios where speed is crucial, and some packet loss is acceptable.
No Connection Establishment: Unlike TCP, UDP does not involve a three-way handshake for connection establishment, resulting in lower latency.
Use Cases in IoT:
Real-Time Applications: UDP is often used in IoT applications where low latency and real-time communication are critical, such as streaming data from sensors.
Broadcast/Multicast: UDP supports broadcast and multicast communication, allowing efficient communication with multiple devices simultaneously.
Relevance in IoT: In IoT, UDP can be used for low-latency applications where real-time communication is more important than guaranteed delivery, such as transmitting sensor data for monitoring or control purposes.
DTLS (Datagram Transport Layer Security):
DTLS is a variant of the Transport Layer Security (TLS) protocol designed for datagram-based communication, such as UDP. It provides encryption, authentication, and data integrity to secure communication channels. DTLS ensures that data is not altered or tampered with during transit.
Characteristics:
Security: DTLS adds a layer of security to UDP, making it suitable for scenarios where secure communication is essential.
Connectionless Security: DTLS provides security features while still maintaining the connectionless nature of UDP.
Use Cases in IoT:
Secure Communication: In IoT applications where data security is a priority, DTLS is used to establish secure communication channels between devices and servers.
Relevance in IoT: DTLS is crucial for securing IoT communication over unreliable networks, where UDP is used. It enables secure transmission of sensitive data and commands between IoT devices and servers while mitigating the risks associated with data loss or interception.

So basically In IoT applications, UDP is often chosen for its low overhead and low-latency characteristics, while DTLS is employed to add a layer of security to UDP communication. This combination allows IoT devices to communicate efficiently while ensuring that the data exchanged is secure and protected against eavesdropping and tampering.

CoAP(Constrained Application Protocol): CoAP or Constrained Application Protocol is a specialized Web Transfer Protocol.CoAP built specially for IOT systems based on HTTP protocol.
CoAP is designed as an alternative to HTTP for IoT applications, providing a simple and low-overhead protocol for machine-to-machine communication.

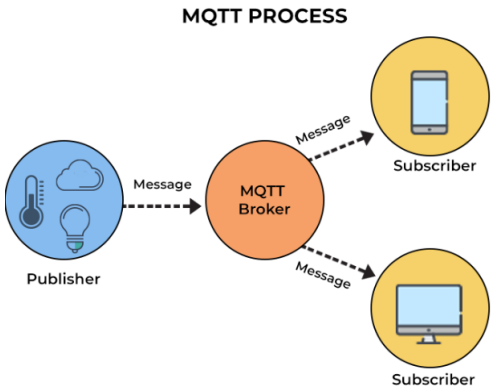
Key Features of CoAP:
RESTful Architecture:CoAP follows a RESTful architecture, making it easy to integrate with web-based technologies and services. It uses similar concepts such as resources, URIs (Uniform Resource Identifiers), and methods (GET, POST, PUT, DELETE).
Lightweight Protocol:CoAP is designed to be lightweight, both in terms of message size and processing requirements. This makes it suitable for resource-constrained devices with limited bandwidth and computational capabilities.
Low Power Consumption:CoAP is designed with low power consumption in mind, making it suitable for battery-operated IoT devices. It minimizes the energy requirements for communication.
Asynchronous Communication:CoAP supports asynchronous communication, allowing devices to exchange messages without waiting for a response immediately. This contributes to a more responsive and efficient communication model.

Similarity of CoAP with HTTP: CoAP Follows the same request-response pattern used by HTTP. The CoAP client sends request to the CoAP server and the server then sends a response back. CoAP uses familiar HTTP features like method(Get, Post, Put, and Delete), URLs and content-type.
Difference in CoAP: CoAP runs on UDP as compares to HTTP, which typically uses TCP. UDP is lighter than TCP. CoAP uses small and reduced set of header(header size is limited to 4 bytes). CoAP supports confirmable and non-confirmable message types.

CoAP vs HTTP:

Feature	CoAP	HTTP
Protocol	It uses UDP.	It uses TCP.
Network layer	It uses IPv6 along with 6LoWPAN.	It uses IP layer.
Multicast support	It supports.	It does not support.
Architecture model	CoAP uses both client-Server & Publish-Subscribe models.	HTTP uses client and server architecture.
Synchronous communication	CoAP does not need this.	HTTP needs this.
Overhead	Less overhead and it is simple.	More overhead compare to CoAP and it is complex.
Application	Designed for resource constrained networking devices such as WSN/IoT/M2M.	Designed for internet devices where there is no issue of any resources.

MQTT(Message Queue Telemetry Transport): MQTT (Message Queue Telemetry Transport) is a popular communication protocol commonly used in the Internet of Things (IoT) ecosystem for exchanging data between devices and applications. It targets large networks of small devices that need to be monitored or controlled from the cloud.
- MQTT is a publish-subscribe based “light weight” messaging protocol for IoT and M2M.
- MQTT uses a broker-based pub-sub architecture in the constrained IoT environment.
- MQTT was introduced by Andy Stanford Clark of IBM and Arlen Nipper of Arcom (now Eurotech) in 1999 and was standardized in 2013.
- It is useful for connections with remote locations where network bandwidth is less.
- As its name states, its main purpose is telemetry, or remote monitoring.
- It targets large networks of small devices that need to be monitored or controlled from the cloud- Numerous applications utilize the MQTT such as health care monitoring, energy meter, and Facebook notification.
- It does not require that both the client and the server establish a connection at the same time.
- It provides faster data transmission, like how WhatsApp/messenger provides a faster delivery. It's a real-time messaging protocol.



Here are the key aspects of MQTT in the context of IoT:

Publish-Subscribe Model: MQTT operates on a publish-subscribe pattern. Devices can publish (send) data to a "topic," and other devices can subscribe (listen) to that topic to receive the data. This allows for efficient distribution of information to interested parties.

Broker-Based Communication: MQTT uses a central server called a "broker" to manage communication between devices. Devices don't communicate directly with each other; instead, they send data to the broker, which then forwards it to the appropriate subscribers.

MQTT Publisher - a sensor or device in IoT world that publishes a piece of information.

MQTT Subscriber – anyone who is interested to subscribe and receive a piece of information they're interested in. It could be a smartphone, a wearable device or even other devices that are interested to know about other devices.

MQTT Broker – An intermediary that receives information from publisher and forwards them to the subscribers.

In summary, MQTT provides a flexible and efficient way for IoT devices to communicate, enabling seamless data exchange, real-time interactions, and efficient use of resources. It's widely used in various IoT applications, including smart homes, industrial automation, healthcare, and more.

Components of the MQTT :

Message: The message is the data that is carried out by the protocol across the network for the application. When the message is transmitted over the network, then the message contains the following parameters:

Payload data, Quality of Service (QoS), Collection of Properties ,Topic Name

Client: In MQTT, the subscriber and publisher are the two roles of a client. The clients subscribe to the topics to publish and receive messages. In simple words, we can say that if any program or device uses an MQTT, then that device is referred to as a client.A device is a client if it opens the network connection to the server, publishes messages that other clients want to see, subscribes to the messages that it is interested in receiving, unsubscribes to the messages that it is not interested in receiving,

In MQTT, the client performs two operations:

Publish: When the client sends the data to the server, then we call this operation as a publish.

Subscribe: When the client receives the data from the server, then we call this operation a subscription.

Server: The device or a program that allows the client to publish the messages and subscribe to the messages. A server accepts the network connection from the client, accepts the messages from the client, processes the subscribe and unsubscribe requests, forwards the application messages to the client

TOPIC: The label provided to the message is checked against the subscription known by the server is known as TOPIC.

MQTT Message Format:

The MQTT uses the command and the command acknowledgment format, which means that each command has an associated acknowledgment. As shown in the above figure that the connect command has connect acknowledgment, subscribe command has subscribe acknowledgment, and publish command has publish acknowledgment. This mechanism is similar to the handshaking mechanism as in TCP protocol.

What is a MQTT payload?

Messages are shared with other devices or software via a broker using MQTT. Every message has a topic, based on which the message can be processed further by the Broker. Additionally, each message contains a message content, the so-called payload. The MQTT payload is not bound to a certain structure and can be designed freely. However, it is helpful to specify a particular structure for the message content so that it can be read by other devices or software. Potential message structures are JSON, XML or OPC UA.

CoAP vs MQTT :

Basis of	COAP	MQTT
Abbreviation	Constrained Application Protocol	Message Queuing Telemetry Transport
Communication Type	It uses Request-Response model.	It uses Publish-Subscribe model
Messaging Mode	This uses both Asynchronous and Synchronous.	This uses only Asynchronous
Transport layer protocol	This mainly uses User Datagram protocol(UDP)	This mainly uses Transmission Control protocol(TCP)
Header size	It has 4 bytes sized header	It has 2 bytes sized header
RESTful based	Yes it uses REST principles	No it does not uses REST principles
Persistence support	It does not has such support	It supports and best used for live data communication
Message Labelling	It provides by adding labels to the messages.	It has no such feature.
Usability/Security	It is used in Utility area networks and has secured mechanism.	It is used in IoT applications and is secure
Effectiveness	Effectiveness in LNN is excellent.	Effectiveness in LNN is low.
Communication Model	Communication model is one-one.	Communication model is many-many.

EXTENSIBLE MESSAGING AND PRESENCE PROTOCOL (XMPP)

Extensible Messaging and Presence Protocol (XMPP), originally known as Jabber, is a communication protocol commonly used for instant messaging and real-time presence tracking. While XMPP was originally designed for instant messaging, its extensibility and decentralized nature have made it relevant in the Internet of Things (IoT) context as well.

Here's how XMPP is used in IoT:

Real-Time Communication: XMPP excels in providing real-time communication between devices and applications. It allows devices to send messages, notifications, and updates to each other in near real-time.

Publish-Subscribe Model: XMPP supports a publish-subscribe model similar to MQTT. Devices can publish information to specific "topics," and other devices can subscribe to these topics to receive updates.

Decentralized Architecture: XMPP follows a decentralized architecture, meaning that devices can communicate directly with each other without relying on a central server. This can be advantageous for IoT scenarios where devices are distributed and connected in a mesh network.

Presence Tracking: XMPP's presence functionality allows devices to indicate their status, such as being online, offline, busy, or available. This is important for IoT devices to know the availability of other devices for communication

Security:XMPP can be secured using Transport Layer Security (TLS) to encrypt communication channels, providing a level of security suitable for IoT applications.

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

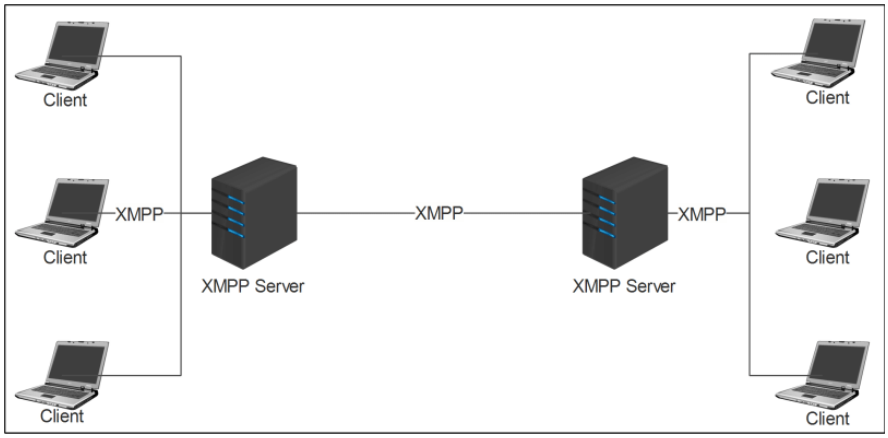
Send and receive messages with other users.

Check and share presence status

Manage subscriptions to and from other users.

Manage contact list

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



NAINA MAM:

IoT Structure:

IoT is a network of tiny innovations like the sensors which can be attached to possibly anything available and then make them communicate with the cloud server without any human interaction.

Sensors - Sensors are devices that collect data from the physical environment. They can measure various parameters such as temperature, humidity, light, motion, and more.

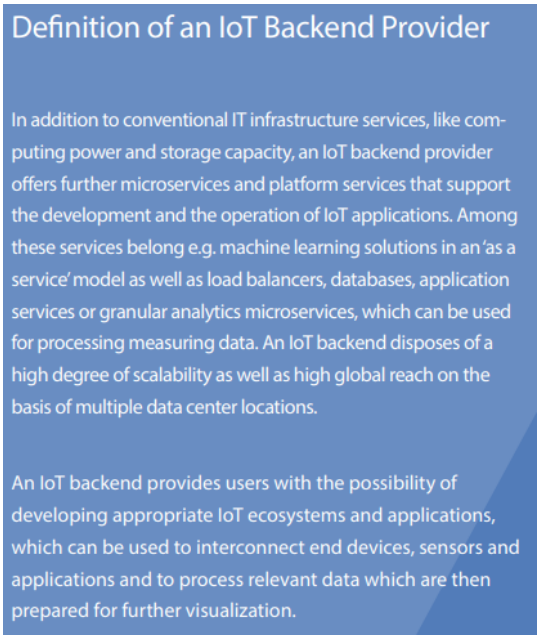
Gateway Device: A gateway device serves as an intermediary between sensors (edge devices) and higher-level networks or cloud services. The gateway device collects data from multiple sensors, processes and analyzes it, and then forwards relevant information to the cloud or central server. It can also perform data filtering, aggregation, and local processing to reduce the amount of data sent to the cloud.

EdgeX Agent: EdgeX is an open-source framework designed for edge computing in IoT systems. An EdgeX agent is a software component that runs on edge devices and facilitates data collection, processing, and communication. EdgeX agent interacts with sensors, collects sensor data, and can apply local rules and analytics. It communicates with the gateway or cloud, helping to manage the flow of data between the edge and higher-level layers.

WebNMS:WebNMS is a management platform that provides solutions for network and service management, as well as IoT device management.n the context of IoT, WebNMS could be used for device monitoring, configuration, and management tasks. It can help track the status of connected devices, monitor their performance, and manage updates and configurations.

Cloud Server: A cloud server is a remote computing environment that provides storage, processing, and networking capabilities over the internet. In an IoT system, the cloud server receives data from sensors, gateways, and edge devices. It can store and analyze large amounts of data, run advanced analytics, generate insights from it.

IOT-Backend-Provisers



Cloud computing: Cloud computing is the on-demand delivery of IT resources over the Internet with pay-as-you-go pricing. OR Computing and software resources that are delivered on demand, as service..

Characteristics of Cloud Computing:

On-demand self-service: On-demand self-service in cloud computing means you can get the computer resources you need, like storage or processing power, whenever you want, without requiring human interaction.

Broad network access: Broad network access in cloud computing means you can use your stuff stored on the cloud from anywhere with the internet, using different devices like your laptop, phone, or tablet. It's like checking your email from any computer – you don't have to be at a specific place to access your things.

Resource pooling: Resource pooling in cloud computing means that the available computer resources, like storage and processing power, are shared and used efficiently among many users. It's like a community pool where everyone can use the water without needing to have their own private pool.

Rapid elasticity: Rapid elasticity in cloud computing is like a stretchy rubber band. You can quickly and easily get more computer power or storage space when you need it, and then shrink it back down when you're done. It's flexible and adjusts to your needs just like a rubber band stretches and contracts.

Measured service / Flexible pricing - Pay per use : Measured service in cloud computing is like paying for your electricity or water usage. You only pay for what you actually use – no more, no less. Similarly, in the cloud, you're charged for the resources you use, such as storage or processing, so it's fair and cost-efficient.

Benefits From Cloud Computing:

Cost: It reduces the huge capital costs of buying hardware and software.

Speed: Resources can be accessed in minutes, typically within a few clicks.

Scalability: We can increase or decrease the requirement of resources according to the business requirements.

Productivity: While using cloud computing, we put less operational effort. We do not need to apply patching, as well as no need to maintain hardware and software. So, in this way, the IT team can be more productive and focus on achieving business goals.

Reliability: Backup and recovery of data are less expensive and very fast for business continuity.

Security: Many cloud vendors offer a broad set of policies, technologies, and controls that strengthen our data security.

Unlimited Storage Capacity.

Device Independence and the “always on!, anywhere and any place”

What is edge computing:

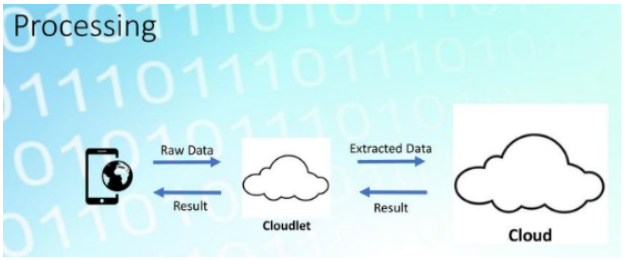
Cloud computing comes with a number of drawbacks. The biggest problem of cloud computing is latency because of the distance between users and the data centers that host the cloud services. This has led to the development of a new technology called edge computing moves computing closer to end users.

Edge computing is a distributed IT architecture which moves computing resources from clouds and data centers as close as possible to the originating source. The main goal of edge computing is to reduce latency requirements while processing data and saving network costs.
The main purpose of edge computing is to reduce latency, improve real-time processing, and enhance efficiency for applications that require immediate data analysis, rapid response time.
Definition: Edge computing refers to the practice of processing and analyzing data closer to the data source or the "edge" of the network, rather than sending all the data to a centralized cloud data center. This approach aims to minimize the latency caused by sending data to distant cloud servers for processing. Instead, edge devices or edge servers process the data locally or in nearby locations, enabling faster decision-making and more efficient resource utilization.

More about edge computing: https://www.fsp-group.com/en/knowledge-app-42.html

CDN: Content Delivery Network (CDN) in the context of the Internet of Things (IoT) refers to a network of geographically distributed servers that work together to deliver content, data, and resources to IoT devices and users efficiently.
Here's how CDNs are relevant to IoT:
Reduced Latency: IoT applications often require low latency to ensure real-time responsiveness. CDNs have servers located closer to end-users and devices, reducing the distance data has to travel and thus decreasing latency.
Load Balancing: CDNs use load balancing techniques to ensure that requests from IoT devices are evenly distributed among the network's servers.
Caching: CDNs cache frequently requested content and data at edge servers.
Scalability: CDNs can dynamically scale to accommodate varying levels of IoT traffic.

Cloudlets: A cloudlet is a lightweight cloud data center that is geographically distributed and located closer to where IoT devices are deployed. It acts as an intermediary between edge devices and the cloud, providing resources for computation, storage, and networking.
By processing data locally on a cloudlet, latency is significantly reduced compared to sending data to a remote cloud. This is important for IoT applications that require real-time or near-real-time response.



Benefits: Reducing Bandwidth cost, improving page load times, increasing global availability of content

Challenges: Higher management cost than centralized infrastructure

Fog Computing: Fog computing is computing that uses devices at the edge of a network (like IoT devices, sensors, or local servers) to do a big part of the computing, storage, and communication work right where they are. Instead of sending everything far away to the central internet, these devices handle a lot of tasks nearby and only send what's necessary over the main internet connection.
CHARACTERISTICS OF THE FOG COMPUTING
Heterogeneity: Heterogeneity in fog computing means that there's a mix of different devices, systems working together.
Real-Time Processing: Fog computing supports immediate data analysis and decision-making, crucial for time-sensitive applications.
Low Latency: Processing data locally or within the network minimizes latency, enabling quicker responses for real-time applications.
Local Decision-Making: Devices at the edge can make decisions locally, reducing the need for constant communication with central servers.
Bandwidth Efficiency: By processing data locally, fog computing reduces the demand on network bandwidth and lowers data transmission costs.
Enhanced Privacy: Sensitive data can be processed locally, reducing the risk of data exposure during transmission.
Support for mobility: It is essential for many Fog applications to communicate directly with mobile devices, and therefore support mobility techniques.

Distributed Architecture: Fog computing involves a distributed network of edge devices, local servers, and gateways that work together to process data.

Designing Goals of Fog Computing:

Latency: It is fundamental for fog computing platform to offer end user low-latency-guaranteed applications and services.

Efficiency: Fog computing aims to utilize resources efficiently, optimizing the use of computing, storage, and network resources at the edge. This involves intelligent task offloading and resource allocation strategies. By processing data closer to the source, unnecessary data transmission to the cloud is minimized, reducing network congestion and conserving bandwidth. Efficient resource utilization also contributes to lower energy consumption

Generality: Due to the heterogeneity of fog node and client, we need provide same abstract to top layer applications and services for fog clients. General application programming interfaces (APIs) should be provided to cope with existing protocols and APIs

Cloud computing vs FOG Computing:

Aspect	Cloud Computing	Fog Computing
Location	Centralized data centers	Distributed computing at the network edge
Data Processing	Heavy processing in data centers	Distributed processing at edge devices
Latency	Can introduce higher latency	Lower latency due to proximity to devices
Network Traffic	Requires significant data transfer	Reduces network traffic by processing locally
Scalability	Scales well with powerful data centers	Scalable with a larger number of devices
Bandwidth Usage	High bandwidth for data transfer	Reduced bandwidth usage by processing locally
Connectivity	Relies on high-speed internet connections	Works with both high-speed and low-speed connections
Device Diversity	Not optimized for a wide variety of devices	Optimized for diverse edge devices
Real-Time Processing	Can be challenging for real-time applications	Well-suited for real-time applications
Data Privacy	Concerns about data leaving edge devices	Data remains closer to source, improving privacy
Resource Utilization	Centralized resources, potential inefficiencies	Efficient use of distributed resources
Resilience to Failures	Can be affected by data center outages	More resilient as processing is distributed
Applications	Well-suited for data analysis, storage	Well-suited for IoT, real-time analytics
Examples	Amazon Web Services, Google Cloud	Cisco IOx, Microsoft Azure IoT Edge

Criteria	Fog Computing	Edge Computing
Location	Closer to the cloud or data center	Closer to the data source or device
Processing	More processing power and storage capacity	Limited processing power and storage
Data Complexity	Handles complex data processing tasks	Handles simpler data processing tasks
Latency	Lower latency due to proximity to cloud	Lower latency due to proximity to devices
Communication	Involves cloud communication	Primarily local communication
Scalability	Supports scalability across regions	Supports scalability at device level
Use Cases	Complex analytics, AI, big data	Real-time data processing, IoT applications
Network Dependency	More dependent on network connection	Less dependent on network connection

IOT gateway: An IoT gateway is a centralized hub that connects IoT devices and sensors to cloud-based computing and data processing.
Data Processing divided into three parts in IoT gateway:
Sensing Domain: The sensing domain is where real-world information is gathered, converted into digital data, and transmitted to the IoT gateway for processing. It refers to the physical world where data is collected by various sensors and devices.
Three sensing domain network :- PAN Network , Vehicle Network , Home Network.
Network Domain: The network domain ensures the efficient and secure transfer of data. Main role is to transfer the data collected from sensing domain to remote destination
Application Domain: The application domain processes the data and turns it into meaningful insights or actions.Take responsibility for Data processing and services.
Working of IoT Gateway :
1.Receives data from sensor network.
2.Performs Pre processing, filtering and cleaning on unfiltered data.
3.Transports into standard protocols for communication.
4.Sends data to cloud.

Common features of IoT gateway:
Protocol Translation: IoT devices often use diverse communication protocols. Gateways facilitate protocol translation, allowing devices with different protocols to communicate effectively.
Data Aggregation: Gateways collect data from multiple edge devices, aggregating it before sending it to the cloud.
Local Processing: Some gateways can perform initial data processing and filtering at the edge. This helps reduce the amount of data sent to the cloud and can enable faster response times.
Connectivity: IoT gateways provide various connectivity options such as Wi-Fi, Ethernet, cellular, Bluetooth, Zigbee, LoRa, and more
Scalability: Gateways can handle multiple connected devices and can scale as the number of devices increases.
Device Management: Gateways often manage connected edge devices, including monitoring their status, updating firmware, and diagnosing issues remotely.

Reference model of IoT gateway

Reference model of IoT gateway

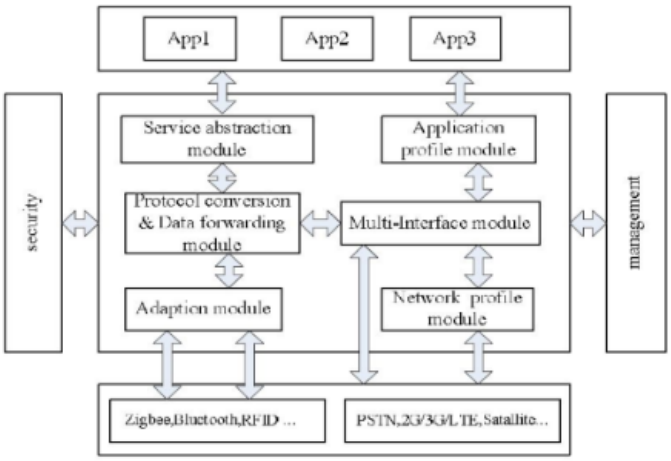


Figure 3 Reference model of IoT gateway

MORE ABOUT IOT GATEWAY & REFERENCE MODEL:
https://drive.google.com/file/d/1BXsRox9yBDkmDWguOl2K0mARFPcSwyy0/view?usp=sharing

TOP CLOUD PLATFORMS: https://drive.google.com/file/d/1Otnu7MH8vqM6bnE8esb-TOScHtd6EWeO/view?usp=sharing

CHAPTER 6: IoT case studies:
Refer Case studies
SMARTGRID: https://drive.google.com/file/d/1q4v9h5kFDvaXh9dXs4L7kF2wT-Pm8lfx/view?usp=sharing

MCQ:

https://drive.google.com/file/d/1KrvzyM-EvnPRLh6ekk0Vr8cx6mWk1mFh/view?usp=sharing

		A	B	C	D	Ans
1	For Iot Application client want to deploy 4,29,49,68,296 device so whiche IP version is use for network configuration ?	IPv4 Only	IPv4 & IPv6 Only	IPv6 Only	Ipv 4 & 6.4 Only	B
2	How to calculet Network ID from IPv6 address ?	Using AND opration	Using AND, OR opration	Using OR opration	None	D
3	6LoWPAN use_____for addressing ?	Stateless Address Auto configuration	Stateless Address	Configurable short addressing	Extended Unique identifier	A
4	Which protocol allowe use to do only the minimum work it needs to transmit data ?	LEACH	LEACH & SPIN	SPIN	MQTT	A
5	What is the time condition to be a cluster head if application use LEACH protocol ?	n < T(n)	n <= T(n)	n > T(n)	n = T(n)	A
6	Which is not anomaly detection methods for software define network ?	Classification-based methods and systems	Clustering and Outlier-based met	Soft computing methods and systems	Flow interruption	D
7	Which is not software define network characteristics ?	Open standards based and Vendor neutral	Centrally managed	Agile Development	None	D
8	Which protocol have lifetime expiry mechanism ?	SPIN	LEACH	AODV 1.1	AODV	D

UNIT TEST PAPER:

https://drive.google.com/file/d/1ux_DrAMPmfcoZplAMEX_tAZzrnsFhojG/view?usp=sharing

OTHER QUESTIONS:

LEVELS OF IOT

https://www.ourtutorials.in/iot/iot_levels.php

Write short note on following terms in perspective of IoT (i) Regulation (ii) Privacy (iii) Standard (iv) Security IN SHORT

(i) Regulation: In IoT, regulation refers to rules and guidelines set by governments or authorities to ensure the safe and ethical use of IoT devices and technologies. These regulations cover aspects such as data privacy, interoperability, spectrum allocation, and safety standards. Adhering to regulations ensures responsible IoT deployment and protects user interests.

(ii) Privacy: Privacy in IoT relates to the protection of individuals' personal information collected by devices. As IoT gathers extensive data, privacy concerns arise. Effective measures must be taken to secure data, obtain user consent, and provide transparency about data usage. Safeguarding privacy maintains user trust and complies with legal requirements.

(iii) Standard: Standards in IoT define common protocols and guidelines for device communication, data formats, and interoperability. Having industry-wide standards ensures seamless integration between devices from different manufacturers and simplifies the development process. Standards enhance compatibility and drive IoT growth.

(iv) Security: Security is paramount in IoT due to the potential risks of unauthorized access, data breaches, and device manipulation. Robust security measures, including encryption, authentication, and regular updates, are essential to safeguard IoT devices and data. Protecting IoT systems prevents vulnerabilities and maintains user confidence.

Write implications of 5'A in IoT.

The "5 A's" in IoT represent five essential components that have significant implications for the success and effectiveness of Internet of Things (IoT) deployments. These components are:

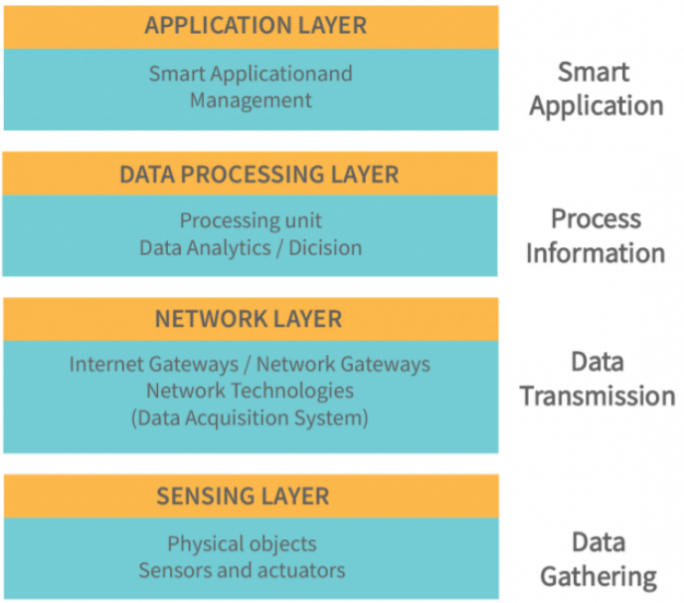
1. Availability:
   - Implication: Reliable and continuous access to IoT devices and data.
   - Ensures devices are operational and responsive when needed.
   - Prevents disruptions and downtime in critical applications.
   - Enables real-time monitoring, control, and data retrieval.
2. Accessibility:
   - Implication: Easy and convenient access to IoT devices and data.
   - Ensures users can interact with devices from various locations and devices.
   - Enhances user experience by providing user-friendly interfaces.
   - Enables remote monitoring and control for improved efficiency.
3. Applicability:
   - Implication: Relevance of IoT solutions to specific use cases.
   - Solutions should address specific business or industry needs.
   - Tailoring IoT technologies to match the requirements of the application.
   - Ensures that IoT solutions provide meaningful insights and value.
4. Affordability:
   - Implication: Cost-effective implementation and operation of IoT solutions.
   - Solutions should offer a reasonable return on investment.
   - Avoids excessive expenses and ensures scalability.
   - Encourages wider adoption of IoT technologies across industries.
5. Acceptability:
   - Implication: User acceptance and willingness to adopt IoT solutions.
   - Solutions should be user-friendly and align with user expectations.
   - Addresses concerns about privacy, security, and data ownership.
   - Increases adoption rates and minimizes resistance to new technologies.

SENSORS AND ACTUATORS

Sensors: https://youtu.be/TtQE3lol6fU?si=JRd1ekj088crsPxt
Classification of Sensors: https://youtu.be/vlgnKDj-9bY?si=mepfDqz6yF_rrqrH
Definition: IoT sensors are pieces of hardware that detect changes in an environment and collect data. They're the pieces of an IoT ecosystem that bridge the digital world to the physical world. IoT sensors may detect things like temperature, pressure, and motion, and if they are connected to a network, they share data with the network.

Actuators: https://youtu.be/omyibPdeU0k?si=bZl7UfvO9phh116K
https://youtu.be/9k4RaKi3v90?si=AnJNz9dDu1i49NEp
Definition: An actuator is a machine component or system that moves or controls the mechanism of the system. Sensors in the device sense the environment, then control signals are generated for the actuators according to the actions needed to perform.
A servo motor is an example of an actuator.
Classification of actuators**:** https://www.codingninjas.com/studio/library/actuators-in-iot
Basically
In IoT applications:
Sensors collect data from the environment, such as temperature sensors in a smart thermostat.
Actuators carry out actions based on the collected data, like adjusting the temperature of a room by controlling a heating or cooling system.
Together, sensors and actuators enable IoT devices to gather information, make decisions, and affect the physical world based on the data they receive. They form the foundation for various applications, from smart homes and industrial automation to healthcare monitoring and environmental sensing.

Architecture of Internet of Things (IoT)



Sensing Layer/ Perception Layer: Devices and sensors gather real word data, includes sensors,actuators and devices.
Network Layer:Sends data from devices to processing units. Uses Wifi, Bluetooth etc.
Processing Later: Processes and analyzes data from devices. Might use edge computing for quicker processing.
Application Layer: Where User interacts with Iot System. Includes apps, dashboards for control and monitoring.

In this simplified diagram:
Data is collected from devices in the Perception Layer.
It moves through the Network Layer to reach the Processing Layer.
Processed data is stored and analyzed in the Processing Layer.
Users interact with the system through the Application Layer

Keep in mind that IoT architectures can vary based on the specific use case and technologies used. This diagram represents a general overview of the components and flow within an IoT system.

Explain the characteristics of IOT. https://youtu.be/nWCd3n6H0as?si=KXiK5esZlo17cirq
https://www.tutorialspoint.com/what-are-the-characteristics-of-internet-of-things-iot

What are different components of IOT.
There are 5 major components of IoT (Internet of Things)
Sensors or Devices: Sensors or Devices are basically used to collect and transmit the data and also perform actions based on those data.For example, the sensors can be used for measuring temperature and humidity.

Gateway: Gateway is also a device component that basically acts as an intermediate between the sensors and the central cloud. Gateway is one of the essential components of IoT that offers communication, management, and data processing.

Cloud: Cloud storage is used to store the data which has been collected from different devices or things. Cloud computing is simply a set of connected servers that operate continuously(24*7) over the Internet.

IoT devices, applications, and users generate massive amounts of data, which must be managed efficiently. Data collection, processing, management, and archiving are among the responsibilities of IoT clouds. The data can be accessed remotely by industries and services, allowing them to take critical decisions at any time.

In the simplest terms, an IoT cloud is a network of servers optimized to handle data at high speeds for a large number of different devices, manage traffic, and analyze data with great accuracy. An IoT cloud would not be complete without a distributed management database system.

Analytics: After receiving the data in the cloud, that data is processed. Data is analyzed here with the help of various algorithms like machine learning and all.

Analytics is the conversion of analog information via connected sensors and devices into actionable insights that can be processed, interpreted, and analyzed in depth. Analysis of raw data or information for further processing is a prerequisite for the monitoring and enhancement of the Internet of things (IoT).

User Interface: User interface also termed as UI is nothing but a user-facing program that allows the user to monitor and manipulate data.

The user interface (UI) is the visible, tangible portion of the IoT device that people can interact with. Developers must provide a well-designed user interface that requires the least amount of effort from users and promotes additional interactions.

System Security: Security is a crucial component of IoT implementation, but this security point of view is too often overlooked during the design process. Day after day weaknesses within IoT are being attacked with evil intent – however, the majority of them that can be easily and inexpensively addressed.

A secure network begins with the elimination of weaknesses within IoT devices as well as the provision of tools to withstand, recognize, and recoup from harmful attacks.

Describe the advantages of IOT.

Minimize human effort: It minimizes human effort because IoT devices connect and communicate with one another and perform a variety of tasks without the need for human intervention.

Save time: By reducing the human effort, it saves a lot of our time. Saving time is one of the primary advantages of using the IoT platform.

Enhanced data collection: Information is easily accessible, even if we are far away from our actual location, and it is updated frequently in real-time. Hence these devices can access information from anywhere at any time on any device.

Improved security: If we have an interconnected system, it can assist in the smarter control of homes and cities through mobile phones. It enhances security and offers personal protection. And Now, if we have a system that all these things are interconnected then we can make the system more secure and efficient.

Useful in the healthcare industry: Patient care can be performed more effectively in real-time without needing a doctor's visit. It gives them the ability to make choices as well as provide evidence-based care.

- It is useful for safety because it senses any potential danger and warns users. For example, GM OnStar, is a integrated device that system which identifies a car crash or accident on road. It immediately makes a call if an accident or crash is found.
- Asset tracking, traffic or transportation tracking, inventory control, delivery, surveillance, individual order tracking, and customer management can all be made more cost-effective with the right tracking system.

Efficient resource utilization: If we know the functionality and the way that how each device work we definitely increase the efficient resource utilization as well as monitor natural resources.

- User Friendly / Easy to Use
- Improving Quality of Life: As IOT increased comfort convenience & better management hence it improves the quality of life.
- Reduce Cost

Disadvantages of IOT:

Lack of Security on privacy - IOT device first share data over the internet, where the risk of losing privacy increases because of hackers.

Complexity - Designing, developing and maintaining IOT is complicated. Any failure or error in the software or hardware will have serious consequences.

- Power failure or no internet connection can cause a lot of convenience.

Increasing Unemployment - As daily activities getting automated there will be fewer requirement of human resources, Specially workers and educated staff.

What are the challenges or risks associated with IOT.

Security and Privacy Concerns: IoT devices can be vulnerable to cyberattacks and hacking due to insufficient security measures.

Power Consumption: Many IoT devices are battery-powered, and optimizing power consumption is crucial for extending device lifetimes and reducing maintenance requirements.

Scalability: As the number of IoT devices grows exponentially, managing and scaling the infrastructure required to support these devices becomes increasingly complex.

Data Management and Analytics: Handling and processing the massive volumes of data generated by IoT devices can be overwhelming.

Cost of Implementation: Developing, deploying, and maintaining IoT systems can be expensive, especially for small and medium-sized businesses or organizations with limited resources.

Lack of Skill and Expertise: The rapid growth of IoT has resulted in a shortage of skilled professionals who understand both the hardware and software aspects of IoT systems.

Overwhelming data volume: The amount of data generated by IoT devices make data oversight, management, and protection difficult.

Which are various OS of IOT.

https://www.javatpoint.com/iot-operating-systems

What are some examples of sensors which can be used in agriculture.

https://www.tractorjunction.com/blog/types-of-smart-sensors-in-agriculture-for-farming-in-india/

What are smart objects. https://youtu.be/tX7K1rdnWKw?si=f-ekKXG_rfTJrCF7

Classification of Smart Objects: https://www.geeksforgeeks.org/classification-of-smart-objects/

What is the impact of IPv6 on IOT.

IPv6 In IoT: IPv6 Is The Most Latest Version Of The Internet Protocol. Devices That Use The Internet Are Recognized By Their Own IP Addresses So That Internet Communication Can Work. IPv6 In IoT Identifies These Devices So That They Can Be Located Through The Internet Easily.

- IPv4 Has 32-Bit Addressing Which Is Able To Support About 4.5 Billion Devices.

But Because Of The Large Number Of Laptops, Computers, Smartphones, And The Internet Of Things Devices, It Was Proved That More Addresses Are Required For The Devices.

- IPv6 Was Created In 1998 Which Uses 128-Bit Addressing. It Supports Approximately 350 Trillion Trillion Devices. The Addressing Method Of IPv6 Includes Eight Groups Of Four Hexadecimal Digits, While The IPv4 Addresses Used To Have Four Sets Of One To Three Digits Numbers.
- The Internet Engineering Task Force(IETF) Who Build IPv4 Decide To Skip IPv5 As It Will Also Eventually Run Out Of Addresses. Hence They Decided To Directly Jump On IPv6 Where There Will Be Nothing To Worry About Running Out Of IP Addresses Again.

Impact: IPv6 has very positive impact on IOT like:

Increased Address Space: The use of IPv6 in IoT devices allows for a significantly larger address space, which is essential for the growing number of devices and sensors that are connected to the internet. This means that the IoT can support a much larger number of connected devices than was previously possible with IPv4.

Efficient Addressing: IPv6 uses a more efficient addressing scheme, which means that it requires less memory and processing power to assign and manage addresses. This makes it more suitable for resource-constrained IoT devices, which often have limited computing resources.

Security: IPv6 offers improved security features, including authentication and encryption, which are essential for securing IoT devices and the data they transmit. With IPv6, IoT devices can securely communicate with each other and with the cloud-based services they rely on.

Auto-configuration: IPv6 devices can automatically configure themselves with unique addresses, which simplifies the process of setting up and managing IoT networks. This is particularly useful in environments where IoT devices are constantly added or moved, as it eliminates the need for manual configuration.

Mobility: IPv6 includes features that support mobile devices, which are becoming increasingly important in the IoT ecosystem. This allows IoT devices to remain connected as they move between different networks, such as home, work, and public networks.

End-to-End Connectivity: IPv6 promotes end-to-end connectivity, allowing IoT devices to communicate directly with each other without the need for intermediary devices or network address translation (NAT). This results in more efficient and direct communication between IoT devices.

Multicasting Support: IPv6 provides native support for multicasting, allowing efficient communication between multiple devices. This is particularly useful in scenarios where one-to-many or many-to-many communication is required, such as in smart cities or industrial IoT applications.

Differentiate between Bluetooth and Bluetooth LE.

S.N.	Features	Bluetooth Classic	Bluetooth Low Energy
1	Frequency Band	2.4 GHZ ISM Band (2.402 GHZ-2.480 GHZ Utilized)	2.4 GHZ ISM Band (2.402 GHZ-2.480 GHZ Utilized)
2	No. of Channels	79 Channel each of width 1 MHZ	40 Channel each of width 2 MHZ
3	Channel Usage/Spreading	Frequency Hopping Spread Spectrum (FHSS)	Frequency Hopping Spread Spectrum (FHSS)
4	Power Consumption	High (Approx. 1W)	Low (Approx. 0.001 W-0.5 W)
5	Communication Range	10m to 30m	10m to 30m
6	Data Rate	1 Mbps for BR 2-3 Mbps for EDR	500kbps-1Mbps
7	Modulation Technique	GFSK for BR 8-DPSK or π/4-DQPSK for EDR	GFSK
8	Communication Direction	Two way directional (Bidirectional)	One way direction (Unidirectional)
9	Device pairing	Required	Not Required
10	Voice capable	Yes	No
11	Latency	100 ms	6 ms
12	Usage cases	Used for streaming applications like audio streaming, file transfer and headsets	Used for sensor data, control of devices, and low bandwidth applications

BLE is used for applications that do not need to exchange large amounts of data, and can therefore run on battery power for years at a cheaper cost.

How edge computing benefits IOT.

Reduced Latency: Edge computing brings processing power closer to the data source, reducing the time it takes for data to travel to a centralized cloud server and back. This is crucial for applications that require real-time or near-real-time processing, such as autonomous vehicles or industrial automation.

Bandwidth Optimization: By processing data closer to where it's generated, edge computing minimizes the need to transmit large amounts of raw data to the cloud. This not only reduces the strain on network bandwidth but also lowers the associated costs.

Enhanced Privacy and Security: Edge computing allows sensitive data to be processed locally, reducing the need to send it to a centralized cloud for analysis. This adds an extra layer of privacy and security, especially important in applications like healthcare or smart homes.

Real-time Decision Making: Applications in IoT often require quick decision-making based on the analyzed data. Edge computing enables devices to make decisions locally without waiting for instructions from a centralized server, leading to faster response times.

Cost Efficiency: Edge computing reduces the need for massive data storage and processing capabilities in centralized cloud servers. This can result in cost savings for organizations, as they can utilize more lightweight and cost-effective edge devices.

Reduced Network Congestion: By processing data locally, edge computing helps in minimizing the amount of data that needs to traverse the network. This reduces network congestion and optimizes overall network performance.

Scalability: Edge computing distributes processing power across a network of devices, allowing for scalable and flexible IoT deployments. As the number of IoT devices grows, edge nodes can be easily added to the network to handle increased processing demands.
Improved Reliability: Edge computing systems can continue to function even when connectivity to the cloud is lost. This ensures that critical operations can still take place locally, providing a more reliable and resilient IoT infrastructure.
Customization and Personalization: Edge computing enables the customization of services based on local data, providing a more personalized and tailored experience for users. This is particularly relevant in applications like retail or smart homes.

What role does a gateway play in IOT.
Gateway provides a bridge between different communication technologies which means we can say that a Gateway acts as a medium to open up connections between the cloud and controller(sensors/devices) in Internet of Things (IoT). With the help of gateways, it is possible to establish device-to-device or device-to-cloud communication.
IoT gateways can be essential in making this connection possible because gateways act as bridges between sensors/devices and the cloud.
Protocol Translation: IoT devices often use diverse communication protocols. Gateways facilitate protocol translation, allowing devices with different protocols to communicate effectively.
Data Aggregation: Gateways collect data from multiple edge devices, aggregating it before sending it to the cloud.
Local Processing: Some gateways can perform initial data processing and filtering at the edge. This helps reduce the amount of data sent to the cloud and can enable faster response times.
Connectivity: IoT gateways provide various connectivity options such as Wi-Fi, Ethernet, cellular, Bluetooth, Zigbee, LoRa, and more
Scalability: Gateways can handle multiple connected devices and can scale as the number of devices increases.
Device Management: Gateways often manage connected edge devices, including monitoring their status, updating firmware, and diagnosing issues remotely.
Security: Gateways play a crucial role in securing the IoT ecosystem. They can implement security measures such as encryption and authentication, safeguarding the data transmitted between devices and the cloud. Gateways act as a barrier, preventing unauthorized access to the IoT network.
Power Management: For battery-powered IoT devices, efficient power management is crucial. Gateways can help optimize power usage by regulating when and how devices transmit data, extending the overall lifespan of battery-powered devices.
Reduce latency: Gateways can reduce latency in time-critical applications by performing processing on the gateway itself rather than in the cloud.

In summary, an IoT gateway serves as a versatile intermediary that enhances communication, security, and management within an IoT ecosystem, contributing to the efficiency and effectiveness of the overall system.

What is IOT edge computing.
Cloud computing comes with a number of drawbacks. The biggest problem of cloud computing is latency because of the distance between users and the data centers that host the cloud services. This has led to the development of a new technology called edge computing moves computing closer to end users.
Edge computing is a distributed IT architecture which moves computing resources from clouds and data centers as close as possible to the originating source. The main goal of edge computing is to reduce latency requirements while processing data and saving network costs.
The main purpose of edge computing is to reduce latency, improve real-time processing, and enhance efficiency for applications that require immediate data analysis, rapid response time.
Definition: Edge computing refers to the practice of processing and analyzing data closer to the data source or the "edge" of the network, rather than sending all the data to a centralized cloud data center. This approach aims to minimize the latency caused by sending data to distant cloud servers for processing. Instead, edge devices or edge servers process the data locally or in nearby locations, enabling faster decision-making and more efficient resource utilization.
https://www.fsp-group.com/en/knowledge-app-42.html

How does fog computing enhance IOT.
Fog computing enhances the Internet of Things (IoT) in several ways by addressing some of the challenges associated with traditional cloud computing. Here are key ways in which fog computing enhances IoT:

Reduced Latency: Fog computing brings computational resources closer to the edge devices, reducing the latency in data transmission. This is crucial for applications that require real-time or near-real-time processing, such as industrial automation, healthcare monitoring, and autonomous vehicles.
Bandwidth Efficiency:By processing data at the edge, only relevant information is sent to the cloud, reducing the amount of data that needs to be transmitted over the network. This is especially important in scenarios where bandwidth is limited or expensive.
Improved Privacy and Security:Local processing of data at the edge reduces the need to send sensitive information to the cloud. This enhances privacy and security by keeping critical data within the local network and minimizing the risk of unauthorized access during data transmission.
Scalability:Fog computing provides a scalable architecture that can adapt to the growing number of IoT devices. Distributing computing resources across the network allows for better scalability, as opposed to relying solely on centralized cloud resources.
Reliability:Edge devices in a fog computing architecture can continue to function even if there is a loss of connectivity to the cloud. This ensures that critical functions can still operate autonomously, even in the absence of a stable internet connection.
Real-time Decision Making:Fog computing enables real-time analytics and decision-making at the edge. This is essential for applications where immediate action is required based on the analyzed data, such as in smart grids or autonomous vehicles.

How data analysis is done using fog computing.
Data analysis in fog computing involves processing and analyzing data at the edge of the network, closer to the source of data generation. Here's a general overview of how data analysis is done using fog computing:
Data Collection at the Edge:IoT devices generate a vast amount of data. In a fog computing architecture, this data is collected at the edge of the network, where the devices are located. This is the first step in the data analysis process.
Preprocessing and Filtering:Before sending the data to the cloud, fog nodes at the edge can perform preprocessing and filtering. This involves cleaning and aggregating the data, as well as filtering out irrelevant or redundant information. This step helps reduce the amount of data that needs to be transmitted to the cloud, optimizing bandwidth usage.
Local Data Analysis:Fog nodes have computational capabilities to perform local data analysis. This could involve running algorithms, machine learning models, or other analytical techniques to extract meaningful insights directly at the edge. Local analysis is particularly useful for applications that require real-time or low-latency responses.
Decision Making at the Edge:Based on the results of local data analysis, fog nodes can make immediate decisions at the edge. This is crucial for applications where timely decisions are required, such as in autonomous vehicles or industrial automation.
Data Aggregation:Aggregated and processed data from multiple edge devices can be further analyzed locally within the fog layer. This allows for a more comprehensive understanding of the overall system or environment.
Selective Data Transmission to the Cloud:Not all data needs to be sent to the cloud for analysis. Fog computing enables selective data transmission, where only relevant or high-priority data is forwarded to the cloud. This optimizes the use of network bandwidth and reduces latency.
Cloud-Based Analysis:For more complex and resource-intensive analysis, fog nodes can forward data to the cloud for further processing. Cloud-based analysis can involve advanced analytics, machine learning, and other data-intensive tasks that may require significant computational resources.

Elaborate fog computing does not replace cloud computing but supports it.
It's a general misconception that Fog computing means "*All computing in the edge* ". That's not true. Fog computing encompasses cloud computing. Fog computing emphasises on a continuum if resources - stretching from enterprise premises (edge) to the data centers (cloud). For tasks with real-time requirements, processing will need to be hosted on the edge; while for those tasks that demand high-end compute will be executed in resourceful data centers. This is what fog computing aims at - placing tasks where they are the most appropriate. I will not call this a replacement, but a **non-trivial extension of cloud computing**.
Apart from it, fog computing requires more infrastructure, which can be expensive to set up and maintain. Additionally, cloud computing is more flexible because it can be used in conjunction with other types of networks. For these reasons, it is unlikely that fog computing will completely replace cloud computing.
Fog computing will never replace the cloud, but will extend it.