**Chapter 1 : Introduction to Embedded Systems**

**1.1)What & where are Embedded systems?**

Embedded systems are **specialized computer systems designed to perform specific tasks.** **They consist of hardware, software, and a real-time operating system (RTOS).** Embedded systems are found in various devices like **microwaves, calculators, and cars.** They have limited processing power compared to regular computers and are programmed using languages like C or assembly. The software for embedded systems is called firmware and is stored in read-only or flash memory.

**1.2 Properties of ES**

1. Embedded systems consume low power to run microcontrollers.

2. They are also of small size in terms of other systems.

3. They have a low cost per unit because of low computing power.

4. It is difficult to program to embed systems and they use intelligent code to operate.

5. Embedded systems are usually made on a large scale so the consumers can get a low-price product.

**1.3 Applications of ES**

Most of the devices we use today are examples of embedded systems. 98% of microprocessors are made for embedded systems.

• They are found in the –

o Consumer electronics: Cell phones, digital camera, Calculator.

o Home Appliances: Microwave oven, washing machine, lightning machine.

o Office Automation: Fax machine, copiers, printers, scanners

o Business Equipment: Alarm system, Card reader, Biometric

**1.4 Characteristics of ES**

**1. Single Functioned**

• An ES usually executes a specific program repeatedly.

• For e.g. pager is always a pager

**2. Tightly Constrained**

• All computing systems have constraints on design metrics but those on ES are tight.

• A design metric is an implementation of features like cost, size, performance and power.

**3. Reactive and Real Time**

• ES must continuously react to changes in the systems environment and must compute results in real time without delay

**1.5 Types of ES**

1. Real-Time Embedded Systems

2. Standalone embedded system

3. Networked embedded system

4. Mobile embedded system

5. Small-scale embedded system

6. Medium scale embedded system

7. Sophisticated embedded system

Embedded systems (ES) can be categorized into different types based on their characteristics, functionalities, and scale. Two common types are:

**1. Real-Time Embedded Systems:**

**These systems are designed to respond to events or inputs within a specific time frame known as a deadline.** Real-time embedded systems are often used in applications where timely and predictable responses are critical, such as in automotive systems (like engine control units), industrial automation (like robotic arms), medical devices (like pacemakers), and aerospace (like flight control systems).

**2.Medium-Scale Embedded Systems:**

This type refers to embedded systems that fall between small-scale and large-scale systems in terms of complexity, processing power, and resource requirements. Medium-scale embedded systems are often found in consumer electronics, appliances, automotive infotainment systems, and some industrial automation applications.

**1.6 Approaches to ES**

**1. General-purpose processors with software:**

These are versatile processors that can run a variety of tasks using software programs.

**2. Single-purpose processors with hardware:**

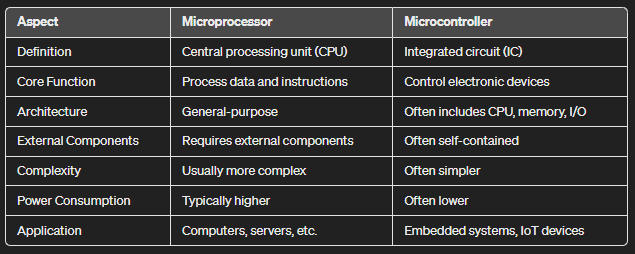
These processors are designed for specific tasks and are built into hardware to perform those tasks efficiently.

**3. Application-specific processors:**

These processors are tailored for specific applications or functions, optimizing performance for those particular tasks.

**1.7 Small Microcontrollers**

**Difference between Microprocessor And Microcontroller**

****

**1.8 Anatomy of a Typical Small Microcontroller**

**Central processing unit:**

• Arithmetic logic unit (ALU), which performs computation.

• Registers needed for the basic operation of the CPU, such as the program counter (PC), stack pointer (SP), and status register (SR)

Memory for the program

Memory for data

Input and output ports

Address and data buses

Clock

Timers

Watchdog Timer

Communication Interfaces

Non-volatile memory for data

Digital-to-analog converter

Real-time clock

Monitor, background debugger, and embedded emulator

**1.9 Memory Architecture**

• Memory lies at the heart of any computer.

• Each location can typically store 1 byte (8 bits or 1B) of data and is called a register.

• Memory is linked to the CPU by buses for data, address, and control.

• Buses are shared sets of wires that join several components.

• Access to the bus must be controlled.

• The job of the control bus is to ensure that the data are valid and two components do not try to write at the same time.

**Volatile memory:**

It is usually called random-access memory or RAM, but the name is misleading because access to most other types of memory is equally random.

• Loses its contents when power is removed.

• The vital feature is that data can be read or written with equal ease.

**Non-Volatile Memory:**

• It retains its contents when power is removed and is therefore used for the program and constant data.

• It is usually called read-only memory or ROM, but again this traditional name has become misleading.

• Most modern microcontrollers can write to their nonvolatile memory but it is much slower and more complicated than writing to RAM.

**There are many types of non-volatile memory in use:**

**i) Masked ROM:**

**ii) EPROM (electrically programmable ROM):**

**iii) OTP (one-time programmable memory):**

**iv) Flash memory:**

**Harvard and von Neumann Architectures**

**1. Harvard Architecture:**

- Uses separate memory spaces for data and instructions.

- Allows fetching instructions and accessing data simultaneously, which can speed up processing.

- Commonly found in microcontrollers and some specialized computing systems.

**2. von Neumann Architecture:**

- Uses a unified memory space for both data and instructions.

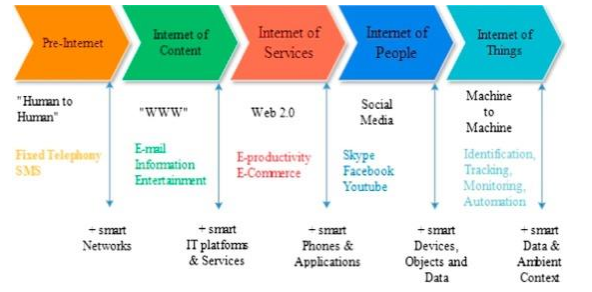
- Instructions and data are fetched sequentially, which can limit processing speed compared to Harvard architecture.

- Most general-purpose computers, including PCs and laptops, follow this architecture.

**Chapter 3: Introduction to IOT**

**3.1 Evolution of the IoT concept**

**The Internet of Things (IoT) concept has evolved over time. Initially, it started with basic machine-to-machine (M2M) communication, where devices could connect and share data without human intervention**. Over the years, IoT has grown to include a wide range of devices, from smartphones and wearables to smart home appliances and industrial machinery. This evolution has been driven by advancements in connectivity, sensor technology, cloud computing, and data analytics, enabling more intelligent and interconnected systems.



**3.2 Vision and Definition of IoT**

**The Internet of Things (IoT) refers to objects like devices and systems that are equipped with sensors, software, and technologies to connect and share data over the internet.**

The vision of IoT includes:

1. Devices that can automatically report real-time data, reducing the need for human intervention and improving efficiency.

2. Objects, such as wearables, alarms, and home devices, become "smart" by sensing, computing, and communicating.

3. Embedded devices interacting with remote objects or people through various technologies like the Internet or Near Field Communication, making connectivity seamless.

**3.3 Basic characteristics of IoT**

1. Scalability

2. Interoperability

3. Real-time Data

4. Automation

5. Connectivity

6. Intelligence and Identity

7. Dynamic and Self-Adapting (Complexity)

8. Architecture

9. Safety

10. Self-Configuring / Upgradation

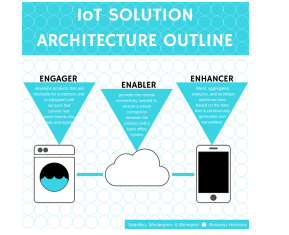
11. Data

12. Sensor Data Acquisition, Storage, Filtering and Analysis

13. Device Heterogeneity and Intelligence

14. Embedded Sensors and Actuators

**3.4 Distinguish the IoT from other related technologies.**

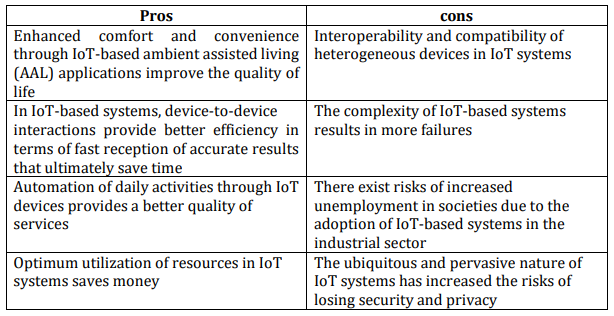
****

**1) Enablers**: Develop and implement the underlying technology.

**2) Engagers:** Design, create, integrate, and deliver IoT services to customers. Engagers offer the direct link between IoT and the market

**3) Enhancers:** Devise their own value-added services on top of the services provided by Engagers.

**3.5 pros and cons of IoT**



**3.6 IOT Challenges & Solutions**

**1. Network**

With the increasing number of IoT devices in a network, it becomes difficult to maintain communication for proper functioning.

**Solution:**

● Cloud service and gateway are a few methods that can solve such problems.

**2. IOT Security**

IoT devices have been vulnerable to cyber-attacks. IoT devices often have a limited power supply and need to last for years in the field on a single charge

**Solution:**

Low-power connectivity solutions continue to implement new security technologies. And this is an area where cellular IoT is particularly valuable

**3. Coverage**

To transmit and receive data, IoT devices need a network connection. Lose the connection, and you lose the device’s capabilities.

**Solution:**

Several technologies provide wide coverage, enabling IoT devices to operate within a few miles of the network infrastructure.

**4. Scalability**

IoT businesses often have hundreds or thousands of devices in the field.

**Solution:**

Global IoT solutions circumvent this challenge by creating agreements with carriers all over the world.

**5. Interoperability**

IoT is the seemingly endless way you can configure your tech stack to suit your unique circumstances.

**Solution:**

Most components of your IoT stack are relatively easy to exchange for other tech.

**6. Bandwidth Availability**

Radio Frequency (RF) bandwidth is a finite resource the entire world has to share.

**7. Limited Battery Life**

**8. Remote Access**

**3.7 IoT architecture**

**IOT architecture comprises 4 layers:**

**Sensing Layer, Network Layer, Data processing Layer, and Application Layer.**

**1. Sensing / Perception Layer**

● The sensing layer is the first layer of the IoT architecture and is responsible for collecting data from different sources.

**2. Connectivity / Network Layer**

● The network layer of an IoT architecture is responsible for providing communication and connectivity between devices in the IoT system.

**3. Edge Or Fog Computing Layer**

The idea behind edge or fog computing is to process and store information as early and as close to its sources as possible.

**4. Data processing Layer**

● The data processing layer of IoT architecture refers to the software and hardware components that are responsible for collecting, analyzing, and interpreting data from IoT devices.

**5. Application Layer**

● The application layer of IoT architecture is the topmost layer that interacts directly with the end-user.

**6. Business Layer**

● The information generated at the previous layers brings value if only it results in problem-solving solutions and achieving business goals.

**7. Security Layer Device security :** Modern manufacturers of IoT devices typically integrate security features both in the hardware and firmware installed on it.

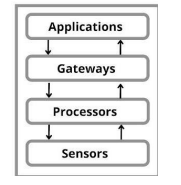
**Connection security :** Whether data is being sent over devices, networks, or applications, it should be encrypted. Otherwise, sensitive information can be read by anybody who intercepts information in transit.

**Cloud security :** Data at rest stored in the cloud must be encrypted as well to mitigate risks of exposing sensitive information to intruders. Cloud security also involves authentication and authorization mechanisms to limit access to the IoT applications.

**Chapter 4: IoT Building Blocks -Hardware and Software**

**4.1 The basic IoT building blocks.**

The basic building blocks of the IoT system are – sensors, processors, gateways, and applications Each of the blocks has its own characteristics to form a dynamic IoT system.



**Sensors / Actuators**

● The sensor is the front end of the IoT system. The main aim of the sensor is to collect data from the environment sensors or give out data to the environment (actuators) from its surroundings (actuators).

**Processors**

● Processors are the brain of the IoT system. The main function is to process the data captured by the sensors and process them so as to extract valuable data from the enormous amount of raw data collected.

**Gateways**

The main function of the gateway is to route the processed data and send it to the location for proper utilization. Examples of a gateway are LAN. WAN PAN. LAN, WAN, PAN, etc are examples of network gateways

**Applications**

● Applications are important for the proper utilization of all the data collected.

**4.2 Basics of Packet Tracer with reference to IoT**

**4.2.1 Cisco Packet Tracer Overview**

Cisco packet tracer is a powerful virtual network simulation tool used to learn and understand different concepts in computer networks. The tool is developed by Cisco to allow students or users to get practical networking technology knowledge.

**4.2.2 Packet tracer Workspaces**

Cisco packet has two Workspaces: one is Physical and the other one logical.

**4.2.3 Packet tracer Mode**

The tool also provides two modes, which are real time mode, and simulation mode.

**4.2.4 Cisco devices configuration methods**

Cisco packet tracer allows us to configure devices using two options: config tab or CLI tab (command line interface).

**4.2.5 Cisco packet tracer supported protocols**

Cisco packet tracer supports different protocols. The table below show the lists of protocols supported by packet tracer

**4.2.6 Cisco packet tracer and Internet of Things**

The last version of Cisco packet tracer included some new features that can help us to perform internet of things simulation. Those new features are smart devices, sensors, actuators and microcontrollers.

**4.3 basics of IoT gateway**

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 the Internet of Things (IoT). With the help of gateways, it is possible to establish device-to-device or device-to cloud communication.

**Key functionalities of IoT Gateway:**

Establishing a communication bridge Provides additional security.

Performs data aggregation.

Pre-processing and filtering of data.

Provides local storage as a cache/ buffer.

Data computing at edge level.

Ability to manage the entire device.

Device diagnostics.

Adding more functional capability.

Verifying protocols

**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 the cloud.

**Advantages of Gateway:**

**Protocol translation:**

IoT devices typically use different communication protocols, and a gateway can translate between these protocols to enable communication between different types of devices

**Data aggregation:** A gateway can collect data from multiple IoT devices and aggregate it into a single stream for easier analysis and management.

**Edge computing**: Gateways can perform edge computing tasks such as data processing, analytics, and machine learning, enabling faster and more efficient decision-making.

**Security:** Gateways can act as a secure access point for IoT devices, providing a layer of protection against cyber threats.

**Scalability:** Gateways can support a large number of IoT devices and can be easily scaled up or down to meet changing needs.

**Improved reliability:** Gateways can help to improve the reliability of IoT devices by managing network connectivity and providing a backup mechanism in case of network failure.

**Cost-effective:** Gateways can be a cost-effective way to manage and control many IoT devices, reducing the need for expensive infrastructure and IT resources

**Chapter 5: Sensing Principles and Wireless Sensor Network**

**5.1.1 What are Sensors?**

**The sensor may be defined as a machine, module, subsystem, or device whose purpose is to detect changes and events in the environment.** The information regarding these changes is further sent to various other electronics mostly consisting of a computer processor. **A sensor is always used along with an electronic device.**

**The applications comprise airplanes and aerospace, medicine, cars, manufacturing and machinery, robotics, and various other aspects related to daily life**

**5.1.2 Classification of Sensors Sensors are classified based on:**

● Output signal.

● Physical parameters measured.

● Output signal (Further classified into analog and digital output sensors)

**5.1.3 Different Types of Sensors with Their Applications**

**1. Temperature Sensors:**

Measure temperature in various applications like HVAC systems, industrial processes, and weather monitoring.

**2. IR Sensors:**

Detects infrared radiation, used in security systems, motion detectors, and temperature measurement.

**3. Ultrasonic Sensor:**

Emit ultrasonic waves and measure their reflection time for distance measurement, object detection in robotics, and parking assistance.

**4. Touch Sensor:**

Detect touch or pressure, commonly found in touchscreens, interactive kiosks, and home appliances.

**5. Proximity Sensor:**

Detect nearby objects without physical contact, used in smartphones for screen dimming, automatic faucets, and robotics.

**6. Chemical Sensor:**

Detect specific chemicals or gases, applied in environmental monitoring, industrial safety, and medical diagnostics.

**7. Gas Sensor:**

Detects gasses like carbon monoxide, methane, or oxygen levels, used in gas leakage detection, air quality monitoring, and automotive applications.

**8. Humidity Sensors:**

Measure moisture levels in the air, used in HVAC systems, weather stations, and food storage.

**9. Acceleration Sensor:**

Measure acceleration forces, found in smartphones for orientation detection, vehicle stability control, and fitness trackers.

**10. Sound Sensor:**

Detects sound levels or vibrations, used in security systems, noise pollution monitoring, and musical instruments.

**45.1.4 Uses of Sensors**

**Sensors are used to detect and measure physical properties such as temperature, light, pressure, motion, and more.**

They are commonly used in various applications such as:

**1. Environmental monitoring:** Sensors can track air quality, water levels, and weather conditions.

**2. Industrial automation:** Sensors are used in manufacturing processes to monitor machinery, control production lines, and ensure safety.

**3. Medical devices:** Sensors are integral to medical equipment like heart rate monitors, blood glucose meters, and imaging devices.

**4. Automotive systems:** Sensors play a crucial role in vehicle safety, monitoring tire pressure, detecting obstacles, and controlling engine performance.

**5. Smart home technology:** Sensors enable automation and control of home devices such as thermostats, lights, and security systems.

**5.2 Applications of Digital Sensors**

**Digital sensors have a wide range of applications across various industries and fields.**

Some common applications include:

**1.Environmental Monitoring:**

Digital sensors are used to monitor environmental parameters such as temperature, humidity, air quality, and pollution levels in real-time.

**2.Industrial Automation:**

They play a crucial role in industrial automation by monitoring parameters like pressure, flow rates, and vibrations in machinery, ensuring optimal performance and safety.

**3.Healthcare:**

Digital sensors are used in medical devices for monitoring vital signs like heart rate, blood pressure, and glucose levels, enabling remote patient monitoring and personalized healthcare.

**4.Smart Homes:**

They are integrated into smart home systems to monitor and control factors such as temperature, lighting, security, and energy consumption for increased convenience and efficiency.

**5. Automotive Industry:**

In vehicles, digital sensors are used for various applications such as monitoring engine performance, detecting obstacles for parking assistance, and enhancing safety through features like collision avoidance systems.

**6.Agriculture:**

Digital sensors help in precision agriculture by monitoring soil moisture, nutrient levels, and environmental conditions, leading to optimized crop yields and resource utilization.

**7. Consumer Electronics:**

They are used in smartphones, wearables, and other electronic devices for functionalities like touch sensing, motion detection, and environmental sensing (e.g., ambient light sensors).

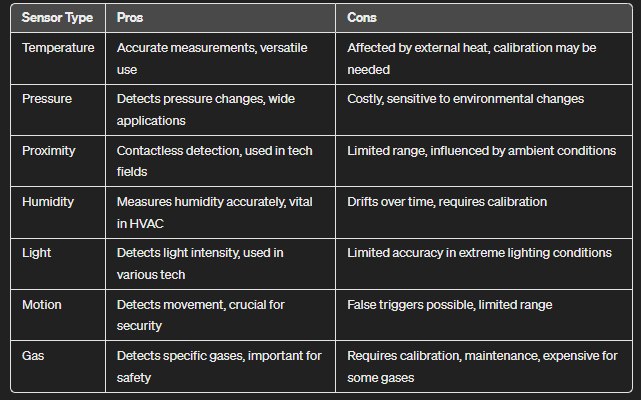
**8.Weather Forecasting**:

Digital sensors contribute data for weather forecasting systems by measuring atmospheric conditions such as temperature, humidity, wind speed, and precipitation.

**5.3 How do Sensors work**

**Sensors work by detecting changes in physical quantities such as light, temperature, pressure, or movement.** They do this by converting these changes into electrical signals that can be interpreted by a computer or other electronic device. For example, a light sensor detects changes in light intensity and converts them into electrical signals that a computer can use to adjust the brightness of a screen or turn on/off a light.

**5.4 Pros and Cons of Different Sensors**



**Chapter 6: IOT Gateway**

**6.1 IoT gateway architecture**

Design of an IoT Gateway is driven by the ‘Custom Application’ , E.g. industrial automation, connected cars.

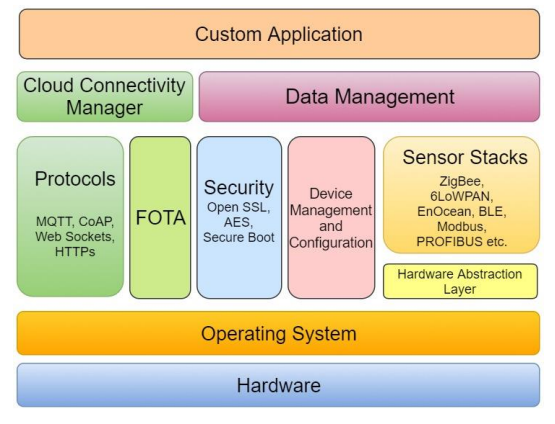
As an IoT developer, based on the requirement of the application one needs to calibrate the following:

IoT Sensors range

Power demands

Performance

Scalability and security



1. IoT Gateway Device Hardware

2. Operating System

3. HAL (Hardware Abstraction Layer)

4. IoT Sensors Stack

5. Device Management and Configuration

6. Security

7. FOTA

8. Data Communication Protocols

9. Data Management

10. Cloud Connectivity Manager

11. Custom Application

12. Gateway Data Transfer

**6.2 IoT gateway functionalities**

1. Communication with the cloud

2. Route the traffic

3. Support of multiple transfer protocol

4. Isolation of sensor nodes

5. Aggregation & processing of data (Edge Computing)

6. Security

7. Local storage of data

**6.3)IoT gateway selection criteria**

1. Connectivity Requirements

2. Data Processing Capabilities

3. Security Features

4. Scalability and Flexibility

5. Power and Energy Efficiency

6. Data Compatibility

7. Connectivity Redundancy

8. Remote Management and Monitoring

9. Environmental Conditions

**6.5 Edge Computing Architectures**

**Pure edge :**

Deploying all compute resources on-premises. This is suitable for organizations with security or compliance requirements that do not allow sending data to the cloud. This requires a larger initial investment.

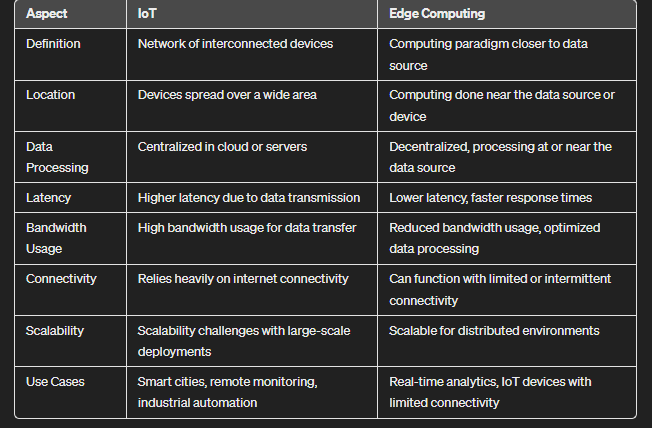
**Thick edge + cloud**:

Deploying an on-premises data center, cloud-based resources, and edge computing devices

**Thin edge + cloud :**

This approach connects edge resources directly to the public cloud, with no on-premise data center.

**6.6 Difference between IOT & Edge computing**

****

**6.7 Edge computing-based solution for specific IoT applications**

**1. Autonomous vehicles**

Autonomous platooning of truck convoys will likely be one of the first use cases for autonomous vehicles.

**2. Oil and gas industry**

Oil and gas failures can be disastrous. Their assets therefore need to be carefully monitored.

**3. Smart Grid**

Edge computing will be a core technology in more widespread adoption of smart grids and can help allow enterprises to better manage their energy consumption

**4. In-hospital patient monitoring**

Healthcare contains several edge opportunities. Currently, monitoring devices (e.g. glucose monitors, health tools and other sensors) are either not connected

**5. Traffic management**

Edge computing can enable more effective city traffic management. Examples of this include optimizing bus frequency given fluctuations in demand, managing the opening and closing of extra lanes, and, in future, managing autonomous car flows.

**6. Smart homes**

Smart homes rely on IoT devices collecting and processing data from around the house.

**Chapter 7: IOT Protocol Stack**

**7.1 IOT Protocol Stack**

**1. Physical layer:**

Responsible for transmitting raw data over a physical medium such as Wi-Fi, Zigbee, NFC, Bluetooth, etc.

**2.Data link layer:**

Provides error correction and flow control mechanisms to ensure reliable data transmission. Protocols like Ethernet, Wi-Fi (802.11), Zigbee (802.15.4), Bluetooth (802.15.1), etc., operate at this layer.

**3.Network layer:**

Handles routing and forwarding of data packets between IoT devices and the Internet or other networks. Protocols like IP (Internet Protocol), IPv6 (Internet Protocol version 6), 6LoWPAN (IPv6 over Low-Power Wireless Personal Area Networks), etc., operate at this layer.

**4.Transport layer:**

Ensures reliable and efficient delivery of data between IoT devices. Protocols like TCP (Transmission Control Protocol) provide reliable, connection-oriented communication, while UDP (User Datagram Protocol) offers connectionless, lightweight communication suitable for real-time applications.

**5. Application layer:**

Provides support for IoT applications and services. Protocols like MQTT (Message Queuing Telemetry Transport), CoAP (Constrained Application Protocol), HTTP (Hypertext Transfer Protocol), MQTT-SN (MQTT for Sensor Networks), AMQP (Advanced Message Queuing Protocol), etc., operate at this layer, facilitating communication and data exchange between IoT devices and applications.

**DNS :**

**The Domain Name System (DNS) is the phonebook of the Internet. When the users type domain names such as ‘google.com’ into web browsers, DNS is responsible for finding the correct IP address for those sites**

Following are the two well-known services discovery protocols used in IOT systems:

**1. Multicast Domain Name System (mDNS)**

Multicast DNS is the DNS equivalent for a local area network. Consider a room that has several devices (mobile phones, laptops, televisions, printers etc.) connected over a common Wi-Fi network.

**2. Domain Name System Services Discovery (DNS SD**)

mDNS can also be used in conjunction with DNS-based Service Discovery (DNS-SD). Most of the devices also provide some services to other devices on the same network.

**Specifications**

The multicast query is always executed on either IPv4 address 224.0.0.51 or IPv6 address ff02 :: fb. It is a UDP message, on port 5353

**Implementation in IOT**

**mDNS (along with DNS-SD) finds applications in discovering smart devices on the local network, based on either the hostname or the service**.

Once discovered, you can configure and interact with the devices over the local network itself.

**7.3 Application layer protocols of IoT protocol stack**

**The application layer acts as an interface between end-user and IoT, the user can utilize the data and manage the device through the set of applications that runs in this layer.**

**1. MQTT (Message Queuing Telemetry Transport):** MQTT sessions have four stages - Connection, Authentication, Communication, and Termination.

**2. SMQTT (Secure MQTT):** It's a secure version of MQTT that ensures encrypted communication.

**3. XMPP (Extensible Messaging and Presence Protocol):** Used for real-time communication and presence information exchange.

**4. AMQP (Advanced Message Queuing Protocol):** Focuses on reliable messaging between devices and systems in IoT applications.

**Chapter 8: IOT Cloud & Fog Computing**

**8.1 Components of IoT Cloud architecture**

**1. IoT Devices:**

Physical devices like sensors, cameras, and actuators that collect data and interact with the environment**.**

**2. IoT Integration Middleware:**

Software that connects IoT devices to the cloud, manages data flow, and enables communication protocols.

**3. Cloud Servers:**

Remote servers hosted in the cloud that store and process IoT data, providing scalability and accessibility.

**4. Databases:**

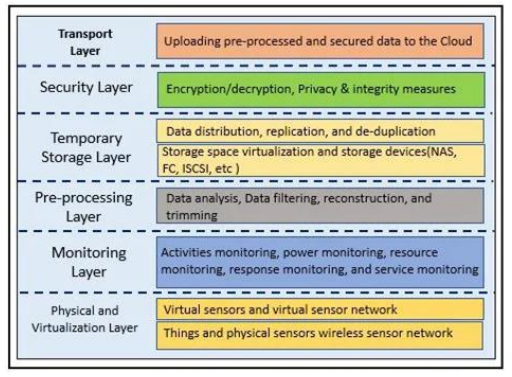
Storage systems in the cloud that organize and manage large volumes of IoT data for analysis and retrieval.

**5. Downstream Applications/BI Tools:**

Software applications and business intelligence tools that utilize IoT data for analysis, visualization, and decision-making.

**8.3 Layered architecture of Fog computing**

**Fog architecture involves using services of end devices (switches, routers, multiplexers, etc) for computational, storage and processing purposes.**



**1. Physical and Virtualization Layer:**

This layer manages the physical hardware resources and virtualization technologies, such as virtual machines or containers.

**2. Monitoring Layer:**

This layer oversees the performance, health, and usage of the system, collecting data for analysis and troubleshooting.

**3. Pre-processing Layer:**

This layer prepares incoming data for further processing, cleaning, transforming, or aggregating it as needed.

**4. Temporary Storage:**

This layer provides temporary storage for processed or intermediate data before it's moved to more permanent storage solutions.

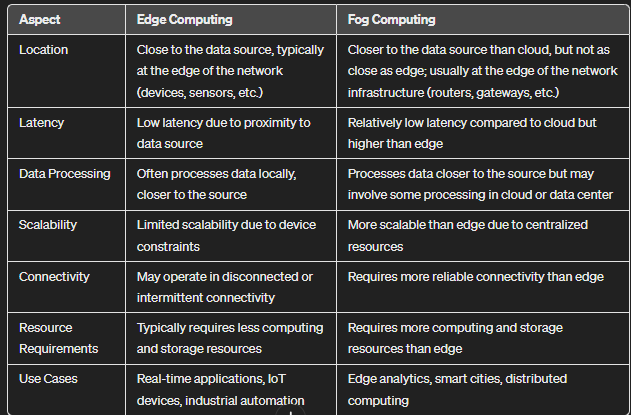
**5. Security Layer:**

This layer ensures the system's security through measures like authentication, authorization, encryption, and firewall protection.

**6. Transport Layer:**

This layer handles the movement of data between different components or systems, ensuring reliable and efficient communication.

**8.4 Difference Between Edge Computing & Fog Computing**

****

**Chapter 9: IOT Applications**

**9.1 Main applications of IoT**

**1. Smart Agriculture:** Using IoT to monitor and manage crops, livestock, and farming conditions for optimized productivity.

**2. IoT Applications in Consumer Use:** Devices like smart home assistants, thermostats, and security systems that enhance convenience and control for consumers.

**3.IoT Applications in Healthcare:** Wearable devices, remote patient monitoring, and smart medical equipment for improved patient care and health management.

**4.IoT Applications in Insurance:** Usage-based insurance models, IoT-enabled risk assessment, and claims processing for the insurance industry.

**5.IoT Applications in Manufacturing:** IoT sensors and automation for efficient production, predictive maintenance, and supply chain optimization.

**6.IoT Applications in Transportation:** Connected vehicles, traffic management systems, and logistics optimization for safer and more efficient transportation.

**7.IoT Applications in Wearables:** Smartwatches, fitness trackers, and health monitoring devices that track personal metrics and provide insights for wellness.

**9.2 Implementation details of various IoT application domains**

**A] Home Automation**

1. Smart Appliances

2. Intrusion detection

3. Smoke Detectors

**B] Smart Cities**

1. Smart Parking

2. Smart Roads

**C] Environments**

1. Air Pollution Monitoring

**D] Retail**

1. Inventory Management

**E] Agriculture**

1. Smart irrigation systems

**F] Health & Lifestyle**

1. Health & Fitness Monitoring

2. Wearable Electronics

**Chapter 10: IOT Security**

**10.1 Security constraints in IoT systems**

1. Lack of encryption

2. Insufficient testing and updating

3. Brute forcing and the risk of default passwords

4. IoT Malware and ransomware

5. IoT botnet aiming at cryptocurrency

6. Inadequate device security

7. Lack of standardization

8. Vulnerability to network attacks

9. Unsecured data transmission

10. Privacy concerns

11. Software vulnerabilities

12. Insider threats

**10.2 Security requirements of IoT systems**

1. Confidentiality

2. Integrity

3. Availability

4. Authenticity

5. Non-repudiation

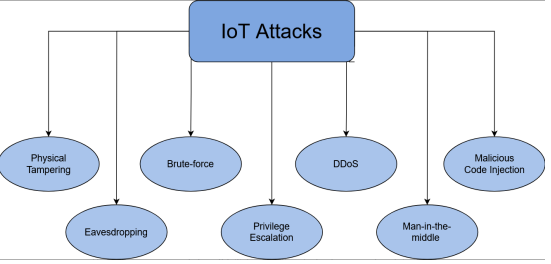
6. Access control and authorization

7. Trustworthy computing

8. Denial-of-Service protection

9. Privacy

**10.3 IoT attacks**

****

**Sure, here are simplified explanations for each of those terms:**

**1. Physical tampering:**

Unauthorized access or modification of physical devices or systems.

**2. Eavesdropping:**

Secretly listening to or monitoring communication between parties.

**3. Brute-force password attacks:**

Trying all possible combinations to guess a password or encryption key.

**4. Privilege escalation:**

Gaining higher levels of access or privileges than originally authorized.

**5. DDoS (Distributed Denial of Service):**

Overloading a system or network with excessive traffic to disrupt its normal functioning.

**6. Man-in-the-middle attack:**

Intercepting and possibly altering communication between two parties without their knowledge.

**7. Malicious code injection:**

Inserting harmful code into a system or application to compromise its integrity or steal data.

**How to prevent IoT attacks?**

1. Use strong, updated passwords.

2. Limit user access to essential personnel.

3. Implement two-factor authentication.

4. Backup data regularly.

5. Encrypt data transmission.

6. Secure device physical access.

7. Restrict user data and device access.

8. Keep software and firmware updated.

9. Conduct regular security audits.

10. Monitor and manage connected devices.

11. Have recovery procedures in place for compromised devices.

**10.4 Security threats at each layer of IoT architecture**

**1. Sensing / Perception Layer Attacks**

Eavesdropping

Node Capture

Fake Node and Malicious

Replay Attack

Timing Attack

**2. Connectivity / Network Layer**

Denial of Service (DoS) Attack

Main-in-The-Middle (MiTM) Attack

Storage Attack

Exploit Attack

**3. Edge / Fog Computing Layer**

Forgery

Tampering

Man-in-the-Middle

**4. Data Processing Layer**

Cross Site Scripting

Malicious Code Attack

The ability of dealing with Mass Data

**5. Application Layer**

Exhaustion

Malwares

**6. Business Layer**

Business Logic Attack

Zero-Day Attack

**10.5 Design secure IoT system for specific application**

**1.Identify Security Requirements:**

Understand the specific security needs of your IoT application. Consider data confidentiality, integrity, authentication, authorization, and availability requirements.

**2.Risk Assessment:**

Conduct a thorough risk assessment to identify potential threats and vulnerabilities. This includes considering physical security, network security, and application-level security risks.

**3.Secure Communication Protocols:**

Use secure communication protocols such as TLS (Transport Layer Security) for data encryption and authentication between IoT devices and the server/cloud.

**4.Authentication and Authorization:**

Implement strong authentication mechanisms (e.g., username/password, certificates) to ensure only authorized users and devices can access the IoT system. Use role-based access control (RBAC) to manage permissions.

**5.Data Encryption:**

Encrypt sensitive data both at rest (stored data) and in transit (data being transmitted) to protect against unauthorized access and data breaches.

**6. Secure Device Management:**

Implement secure device onboarding and management processes. This includes securely provisioning devices, updating firmware/software securely, and revoking access for compromised or unauthorized devices.

**7. Monitoring and Logging:**

Implement continuous monitoring of IoT devices, networks, and applications for any suspicious activities or anomalies. Maintain logs for auditing and forensic analysis in case of security incidents.

**8. Physical Security Measures:**

Implement physical security measures to protect IoT devices from tampering or unauthorized access. This may include secure installation, tamper-resistant hardware, and access control measures.

**Chapter 11: Social IOT**

**11.1 What is Social IOT?**

**Social IoT (SIoT) refers to a segment of the Internet of Things (IoT) that focuses on enabling objects to form social connections with each other based on human interactions.** SIoT aims to address IoT challenges like scalability, trust, and resource discovery by adopting concepts from social computing.

**11.2 Basic components of SIoT**

**Sure, here's a simplified answer to the question about the basic components of SIoT (Smart Internet of Things):**

**1.ID:**

Unique identifiers like MACID, IPv6ID, or custom methods that help identify objects/devices in the system.

**2.Meta-information:**

Describes the form and functions of a device, enabling it to interact with other devices effectively.

**3.Security controls:**

Similar to a friend list on social media, owners can set restrictions on how devices connect to their system.

**4.Service discovery:**

Dedicated directories storing device details and services offered, allowing devices to find and connect with each other.

**5. Relationship management:**

Managing connections and interactions between devices, such as between a light controller and a sensor.

**6.Service composition:**

Integrating services for better user experience, like analyzing data patterns for improved outputs or services.

**11.3 Types of relationships in SIOT**

1. C-WOR (Co-Work Object Relationship)

2. OOR (Ownership Object Relationship)

3. POR (Parental Object Relationship)

4. C-LOR (Co-Location Object Relationship)

5. SOR (Social Object Relationship)

**11.4 Social IoT Architecture**

1. ID Management (ID)

2. Object Profiling (OP)

3. Owner Control (OC)

4. Relationship Management (RM)

5. Service Discovery (SD)

6. Service Composition (SC)

7. Trustworthiness Management (TM)

**11.5 Social IoT Applications**

1. Smart Retailing

2. Smart Traffic Management and Surveillance System

3. Smart Healthcare

4. Smart Shoe

5. Smart Museum/Public Place

**11.6 Advantages of Social IoT**

**Here are the advantages of Social IoT (SIoT) in a simplified manner:**

**1. Navigability:**

SIoT is structured logically, making objects easily discoverable and scalable like a social network, enhancing user experience.

**2. Increased Security:**

SIoT improves security and trust by allowing smart objects to distinguish known and unknown devices, adjusting interaction levels accordingly. This enables better monitoring of security parameters by humans.

**3.Social Connectivity:**

SIoT facilitates objects to establish relationships autonomously, reducing the need for human intervention and enhancing connectivity among devices.