

INTERNET OF THINGS (IoT)

CTEVT — Diploma in IT / Computer Engineering

Year/Part: III/II

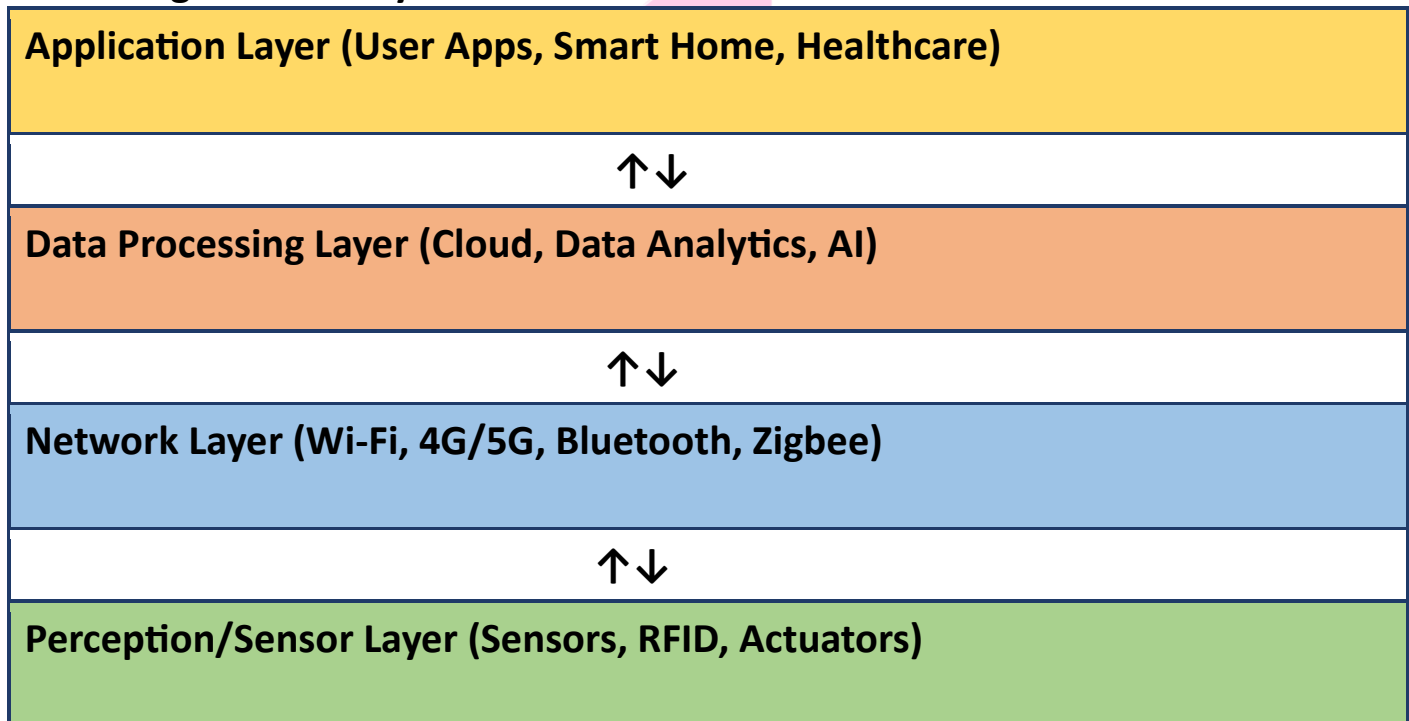
Exam Year: 2082 Shrawan/Bhadra

Q1. What are the four layers of IoT architecture? What is the primary function of each layer? [5+5]

IoT Architecture — 4 Layers:

IoT architecture is a framework that shows how different components of IoT work together. The most common model has 4 layers, each with its own job:

Block Diagram — 4-Layer IoT Architecture:



Function of Each Layer:

1. Perception Layer (Sensor Layer): This is the lowest layer. Its main job is to collect data from the physical world using sensors (temperature, humidity, motion), RFID tags, and actuators. It converts real-world information into digital signals.

2. Network Layer: It transfers the collected data from sensors to the processing systems through the internet. It uses Wi-Fi, Bluetooth, Zigbee, 4G/5G, and other communication technologies.

3. Data Processing Layer (Middleware Layer): This layer receives raw data from network layer, stores it in databases/cloud, and processes it using AI and data analytics tools to extract useful information.

4. Application Layer: This is the top layer that delivers services to users. Examples: Smart home apps, healthcare monitoring, smart agriculture, and industrial automation.

Page

Q2. Explain the main environmental characteristics that impact the performance of IoT systems. [10]

Environmental Characteristics Affecting IoT Performance:

IoT devices work in different environments, so various factors affect how well they perform. Main environmental characteristics are:

1. Temperature: Extreme heat or cold can damage sensors and batteries. Devices need to work in wide temperature range.

2. Humidity: High moisture can cause corrosion and short circuits in electronic devices.

3. Dust and Dirt: Dust particles can block sensors and reduce their accuracy. Industrial IoT devices need dust-proof design.

4. Vibration and Shock: IoT devices in vehicles or machines face vibrations that can damage internal components.

5. Electromagnetic Interference (EMI): Other electronic signals can disturb IoT wireless communication.

6. Power Availability: Remote areas may have unstable power. IoT devices need battery backup or solar power.

7. Network Connectivity: Weak Wi-Fi or mobile signals reduce data transmission speed and reliability.

8. Light Conditions: Some sensors like cameras and light sensors get affected by too much or too little light.

9. Water Exposure: Rain or water can damage devices; waterproof design is needed for outdoor IoT.

10. Physical Security: Devices in open areas can be stolen or tampered with, affecting system performance.

Page

Q3. What are the key technologies that enable the functioning of IoT systems?

Key Technologies of IoT:

IoT works with a combination of many technologies. Below are the main enabling technologies:

- 1. Sensors and Actuators:** Sensors collect data (temperature, motion, light), and actuators perform actions (turn on fan, open door).
- 2. RFID (Radio Frequency Identification):** Uses radio waves to identify and track objects using tags and readers. Used in logistics and inventory.
- 3. Wireless Sensor Networks (WSN):** Group of sensor nodes that communicate wirelessly to collect and share data.
- 4. Cloud Computing:** Provides storage and processing power over the internet to handle huge IoT data.
- 5. Big Data Analytics:** Analyzes massive amounts of IoT data to find useful patterns and insights.
- 6. Artificial Intelligence (AI) and Machine Learning:** Helps IoT devices learn from data and make smart decisions automatically.
- 7. Communication Protocols:** Wi-Fi, Bluetooth, Zigbee, LoRa, MQTT, CoAP are used for device communication.
- 8. Embedded Systems:** Small computers (like Arduino, Raspberry Pi) built into IoT devices.
- 9. GPS and Location Technology:** Provides location tracking for vehicles, animals, and mobile IoT devices.
- 10. Cybersecurity Technology:** Encryption, authentication, and firewalls protect IoT data and devices.

Page

Q4. Differentiate between M2M and WSN. List respective protocols. [8+2]

M2M vs WSN — Difference Table:

Aspect	M2M	WSN
Full Form	Machine-to-Machine Communication	Wireless Sensor Network
Definition	Direct communication between two machines without human help.	Network of sensor nodes that collect and send data wirelessly.
Focus	Communication between machines.	Sensing and data collection from environment.

Devices	Machines, meters, industrial equipment.	Small sensor nodes with limited power.
Power Usage	May use continuous power supply.	Low power (battery-based).
Communication	Point-to-point (one machine to another).	Multi-hop (data passes through many nodes).
Range	Long distance possible (via cellular).	Short distance mostly (10m to few km).
Data Type	Structured machine data.	Sensor readings (temperature, humidity).
Cost	Higher cost.	Lower cost per node.
Example	Smart meters, ATM to bank server.	Forest fire detection, weather stations.

Protocols Used:

M2M Protocols:

MQTT

CoAP

HTTP/HTTPS

OMA-DM

3GPP

LWM2M

WSN Protocols:

Zigbee

6LoWPAN

IEEE 802.15.4

LEACH

SPIN

Bluetooth LE

Page

Anjad Ali
PUBLISHER

Q5. Explain Python programming language. Why is it a good programming language for Raspberry Pi? [4+6]

Python Programming Language:

Python is a high-level, interpreted, object-oriented programming language known for its simple syntax and easy readability. It was created by Guido van Rossum in 1991 and is widely used in web development, data science, AI, and IoT.

Features of Python:

Simple and easy to learn.

Free and open-source.

Cross-platform (works on Windows, Linux, Mac).

Large collection of libraries.

Supports OOP concepts.

Dynamically typed language.

Why Python is Good for Raspberry Pi:

1. Pre-installed on Raspberry Pi OS: Raspberry Pi comes with Python already installed, so no setup needed.

2. Easy Syntax: Beginners can quickly learn and write IoT programs in Python.

3. GPIO Library Support: RPi.GPIO library lets you control Raspberry Pi's GPIO pins to connect sensors and LEDs.

4. Huge Library Support: Libraries for sensors, cameras, motors, and AI make development fast.

5. Community Support: Big community of developers to help beginners.

6. Rapid Prototyping: Quick to write and test small IoT projects.

7. Integration with Cloud: Easy to connect Raspberry Pi with AWS, Google Cloud using Python.

8. Supports AI/ML: Can run TensorFlow Lite and OpenCV on Raspberry Pi for smart IoT.

Page

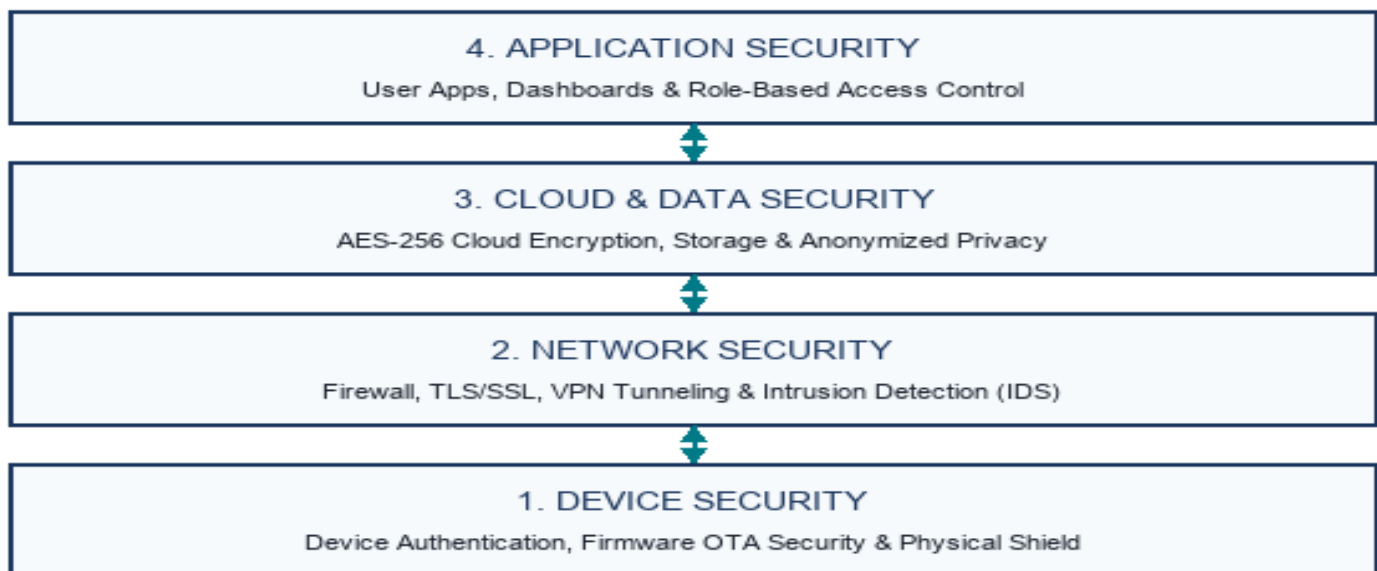
Q6. What are the key components of an IoT security model? [10]

IoT Security Model — Key Components:

IoT security model protects IoT devices, network, and data from cyber attacks. It has these main components:

- 1. Device Authentication:** Every IoT device must prove its identity before joining the network (using digital certificates, passwords).
- 2. Data Encryption:** Data is converted into unreadable form during transmission using AES, RSA, TLS/SSL protocols.
- 3. Access Control:** Only authorized users can access IoT devices and data (role-based access).
- 4. Secure Communication:** Uses HTTPS, TLS, and secure protocols to protect data flow between devices.
- 5. Firewall:** Blocks unauthorized traffic between IoT network and internet.
- 6. Intrusion Detection System (IDS):** Monitors network activity and alerts on suspicious behavior.
- 7. Firmware Update Security:** Devices should get secure updates to fix security bugs.
- 8. Physical Security:** Protects IoT devices from being stolen or tampered with physically.
- 9. Data Privacy:** Personal user data is anonymized and stored securely.
- 10. Security Monitoring & Logging:** Keeps records of all activities for future audit and forensics.

Block Diagram — IoT Security Model:



IOT SECURITY MODEL - 4-TIER ARCHITECTURE BLOCK DIAGRAM

Q7. What are the roles of data analytics in IoT applications? How does data analysis enhance functionality and efficiency in IoT system? [4+6]

Role of Data Analytics in IoT:

Data analytics is the process of examining IoT data to find patterns, insights, and useful information that helps businesses make smart decisions.

Real-time Monitoring: Analyzes live data from sensors to give instant updates.

Predictive Maintenance: Predicts machine failure before it happens, saving cost.

Decision Making: Provides data-based insights for better business decisions.

Improving User Experience: Personalized recommendations based on user behavior.

How Data Analysis Enhances IoT Functionality & Efficiency:

1. Faster Response Time: Real-time analytics allows IoT systems to respond immediately (e.g., auto-braking cars).

2. Predictive Maintenance: Predicts equipment failures early and reduces downtime.

3. Energy Efficiency: Optimizes power usage in smart homes and factories.

4. Better Resource Management: Helps in efficient use of water, electricity, and manpower.

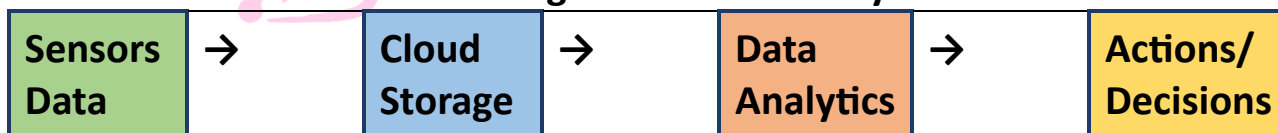
5. Automated Actions: Systems can take auto-decisions like turning off AC when no one is in room.

6. Business Insights: Companies get insights about customer preferences and market trends.

7. Enhanced Security: Detects abnormal patterns and cyber threats in real-time.

8. Cost Reduction: Reduces operational cost through optimization.

Block Diagram — Data Analytics in IoT:



Q8. Describe the role of APIs in IoT. [10]

API (Definition):

API (Application Programming Interface) is a set of rules and tools that allows different software applications and devices to communicate with each other. In IoT, APIs help devices, cloud, and apps exchange data.

Role of APIs in IoT:

- 1. Device-to-Cloud Communication:** APIs let IoT devices send data to cloud servers for storage and processing.
- 2. Data Sharing:** APIs help different apps share IoT data with each other securely.
- 3. Integration with Third-party Services:** IoT systems can connect with services like Google Maps, weather APIs, payment gateways.
- 4. Remote Control:** Users can control IoT devices from mobile apps through APIs (e.g., turn on smart light).
- 5. Automation:** APIs allow IoT systems to trigger auto-actions (e.g., IFTTT recipes).

Common IoT API Types:

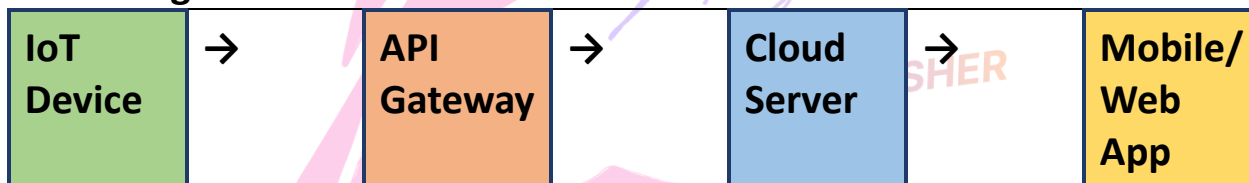
REST API — most common, uses HTTP.

MQTT API — lightweight for IoT devices.

WebSocket API — real-time two-way communication.

CoAP API — for constrained devices.

Block Diagram — API in IoT:



Q9. Write short notes on: (any TWO) [2×5]

a) RFID Technology:

RFID (Radio Frequency Identification) is a wireless technology that uses radio waves to identify and track objects using RFID tags and RFID readers.

Components:

RFID Tag — chip attached to object.

RFID Reader — reads data from tag.

Antenna — transmits radio signals.

Backend System — stores and processes data.

Uses:

Inventory tracking in shops.

Toll collection on highways.

Library book tracking.

Access control cards.

Animal tracking.

b) Satellite Technology:

Satellite technology uses artificial satellites in space to send and receive signals for communication, navigation, and observation. In IoT, satellites help connect remote areas where mobile/Wi-Fi does not reach.

Uses in IoT:

GPS for vehicle and asset tracking.

Remote agriculture monitoring.

Weather forecasting.

Marine and aviation tracking.

Disaster management in remote areas.

c) Vulnerabilities of IoT:

IoT vulnerabilities are weak points that hackers can exploit to attack IoT devices or networks.

Common Vulnerabilities:

Weak Passwords: Default passwords are easy to guess.

No Encryption: Data sent in plain text can be stolen.

Outdated Firmware: Old firmware has known bugs.

Insecure APIs: Weak APIs allow unauthorized access.

Lack of Physical Security: Devices can be stolen or tampered.

Poor Network Security: Open Wi-Fi allows hackers to enter.

d) IoT Devices:

IoT devices are smart physical objects with sensors, software, and internet connection that can collect, exchange, and process data automatically.

Examples:

Smart Home: Amazon Echo, Google Nest, smart bulbs.

Wearables: Fitness bands, smart watches.

Healthcare: Smart glucose monitor, ECG patches.

Industrial: Smart meters, industrial sensors.

Vehicles: Connected cars, GPS trackers.

Agriculture: Soil moisture sensors, drones.

Page

Exam Year: 2081 Kartik/Mangsir

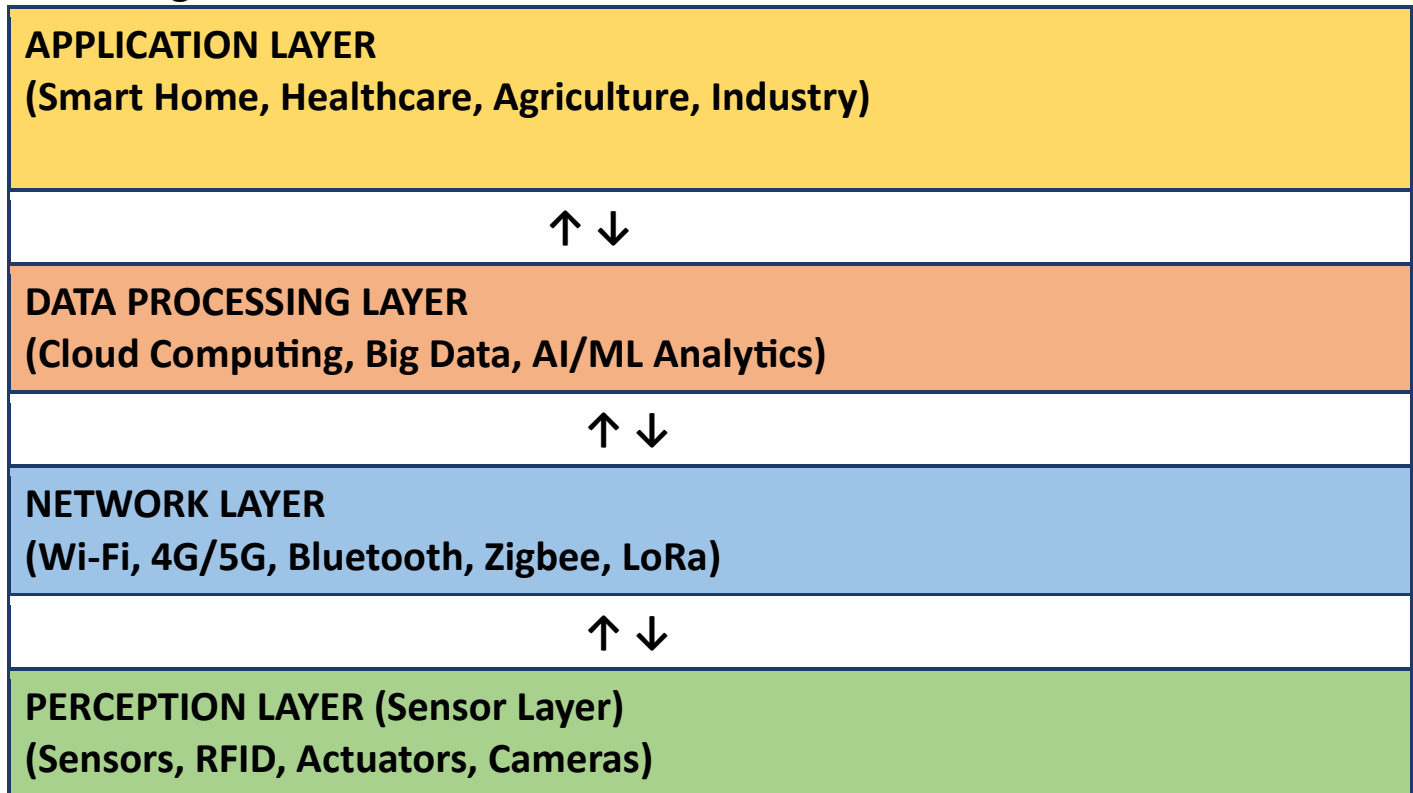
Q1. What is Internet of Things? Explain the IoT architecture with diagram. [2+6]

Internet of Things (IoT) is a network of physical objects (things) like sensors, vehicles, appliances, and machines that are connected to the internet to collect, share, and exchange data with each other without human involvement.

IoT Architecture:

IoT architecture is a structured framework that shows how different components of IoT work together. It usually has 4 main layers:

Block Diagram — IoT Architecture:



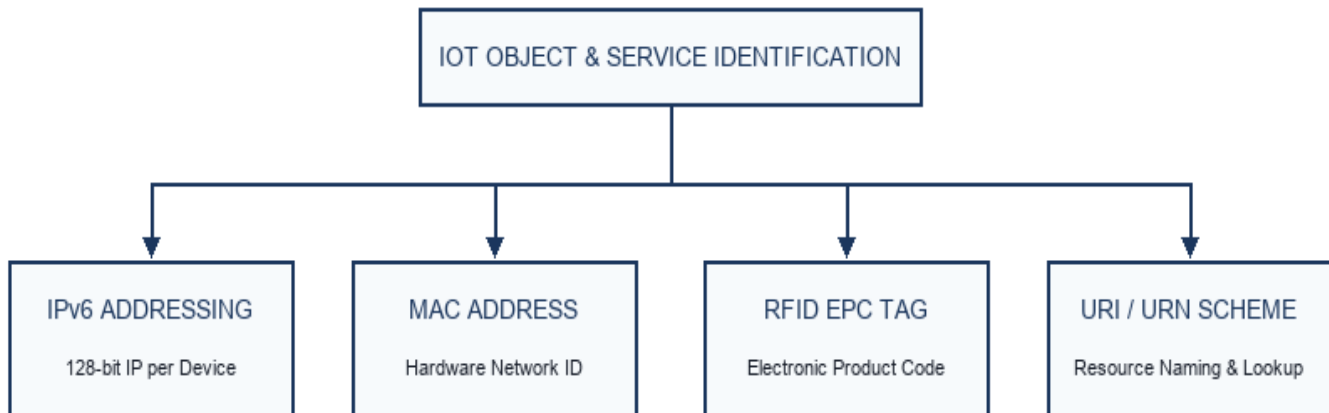
Explanation of Layers:

- **Perception Layer:** Collects data from environment using sensors and RFID.
- **Network Layer:** Transfers collected data to processing systems via internet.
- **Data Processing Layer:** Stores and analyzes data using cloud and AI.
- **Application Layer:** Delivers services to users through apps (smart home, healthcare).

Q2. Discuss the importance of identifying IoT objects and services. How does proper identification contribute to the efficiency and security of IoT systems? /

Importance of Identifying IoT Objects and Services:

In IoT, millions of devices are connected. To manage them properly, each device (object) and its service must have a unique identity. This is called object/service identification.



PROPER IDENTIFICATION ENHANCES EFFICIENT ROUTING & SECURITY

- 1. Unique Recognition:** Every IoT device gets a unique ID (like IP address, MAC address) so it can be recognized in network.
- 2. Better Management:** Admins can easily monitor, control, and update devices when they are identified properly.
- 3. Data Accuracy:** Data from each device is correctly labeled — no confusion between devices.
- 4. Efficient Communication:** Devices know where to send data based on unique identifiers..

Contribution to Efficiency & Security:

- 1. Precise Data Routing:** Ensures sensor data reaches the exact target cloud server without packet loss.
- 2. Access Control & Security:** Blocks unauthorized rogue devices from spoofing network IPs.
- 3. Service Discovery:** Allows applications to locate and communicate with specific sensor services dynamically.
- 4. Global Scalability:** IPv6 provides trillions of unique IP addresses for massive global IoT expansion.

Q3. Evaluate the role of key IoT technologies such as sensor technology, RFID and satellite in enhancing IoT applications. [8]

Sensor Technology:

Sensors are devices that detect physical changes (temperature, light, motion) and convert them into electrical signals for IoT systems.

Role:

Collects real-time data from environment.
Enables smart automation (e.g., auto lights in rooms).
Used in healthcare, agriculture, home, industry.
Provides input for AI/ML systems.
Detects hazards (fire, gas leaks, floods).

2. RFID (Radio Frequency Identification):

RFID uses radio waves to identify and track objects using RFID tags and readers.

Role:

Tracks products in supply chain and warehouses.
Used in toll systems and access cards.
Provides unique ID to every object.
Automates inventory management.
Reduces manual work and errors.

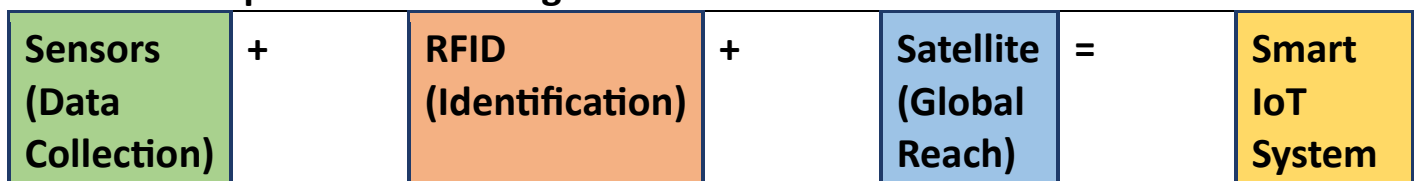
3. Satellite Technology:

Satellites in space help IoT devices communicate over long distances, especially in remote and hard-to-reach areas.

Role:

Provides GPS location tracking.
Connects IoT in remote areas (mountains, oceans).
Helps in weather monitoring and disaster warning.
Supports maritime and aviation IoT.
Enables global connectivity for IoT.

Combined Impact — Block Diagram:



Q4. Explain the importance of IoT protocol standardization. Discuss in brief about SCADA and BACnet. [2+6]

Importance of IoT Protocol Standardization:

Standardization means creating common rules and formats for IoT communication so that different devices from different companies can work together smoothly.

- **Interoperability:** Devices from different brands can communicate with each other.
- **Scalability:** Easy to add new devices without compatibility issues.
- **Reduced Cost:** Standard parts and protocols reduce development cost.
- **Better Security:** Standard security rules protect all devices equally.
- **Wider Adoption:** Standard protocols encourage more companies to develop IoT products.
- **Faster Development:** Developers reuse standard code and libraries.

SCADA (Supervisory Control and Data Acquisition):

SCADA is a system used in industries to monitor and control large-scale operations like power plants, water supply, and manufacturing. It collects real-time data from sensors and controls machines remotely.

Features of SCADA:

Real-time monitoring of industrial processes.

Remote control of machines.

Data logging and reporting.

Alarm system for abnormal events.

Used in power, water, oil & gas industries.

BACnet (Building Automation and Control Network):

BACnet is a communication protocol designed for building automation systems like HVAC (heating, ventilation, AC), lighting, fire alarms, and security in smart buildings.

Features of BACnet:

Open standard protocol for buildings.

Controls HVAC, lighting, and security.

Works over Ethernet, Wi-Fi, MS/TP networks.

Interoperable across different vendors.

Widely used in smart building IoT.

Page

Q5. Differentiate between M2M and WSN protocol. [8]

M2M vs WSN Protocol Comparison:

Aspect	M2M Protocol	WSN Protocol
Full Form	Machine-to-Machine communication protocol.	Wireless Sensor Network protocol.
Purpose	Communication between two machines.	Data collection from sensor nodes.
Devices	Industrial machines, meters.	Small low-power sensor nodes.
Power	Continuous power supply.	Battery-operated (low power).
Range	Long range (cellular based).	Short range (10m - few km).
Topology	Point-to-point.	Mesh or multi-hop.
Protocols	MQTT, CoAP, HTTP, LWM2M.	Zigbee, 6LoWPAN, IEEE 802.15.4.
Data Rate	Higher data rate.	Low data rate.
Cost	Higher cost per node.	Lower cost per node.
Examples	Smart meters, ATM machines.	Weather stations, forest monitoring.

Q6. Describe the process of building an IoT system using Raspberry Pi. What are key components and steps involved? [8]

Raspberry Pi:

Raspberry Pi is a small, low-cost single-board computer that can be used to build IoT projects. It has a processor, RAM, GPIO pins, USB ports, HDMI, Wi-Fi, and Bluetooth.

Key Components Required:

- 1. Raspberry Pi Board:** Main computer (e.g., Raspberry Pi 4).
- 2. SD Card (with Raspberry Pi OS):** Storage for operating system and files.
- 3. Power Supply:** 5V USB-C power adapter.
- 4. Sensors:** Temperature, motion, humidity sensors, etc.
- 5. Actuators:** Motors, LEDs, relays.
- 6. Jumper Wires & Breadboard:** For circuit connections.
- 7. Internet Connection:** Wi-Fi or Ethernet cable.
- 8. Programming Language:** Python (mostly used).
- 9. Cloud Platform:** AWS, Google Cloud, ThingSpeak for data storage.
- 10. Monitor/Keyboard/Mouse:** For initial setup (optional after headless setup).

Steps to Build IoT System with Raspberry Pi:

Step 1: Install Raspberry Pi OS on the SD card using Raspberry Pi Imager.

Step 2: Boot up Raspberry Pi and configure Wi-Fi, SSH, and updates.

Step 3: Connect sensors and actuators to GPIO pins using jumper wires.

Step 4: Write Python program using RPi.GPIO library to read sensor data.

Step 5: Process the data locally on Raspberry Pi.

Step 6: Send data to cloud platform using MQTT or HTTP API.

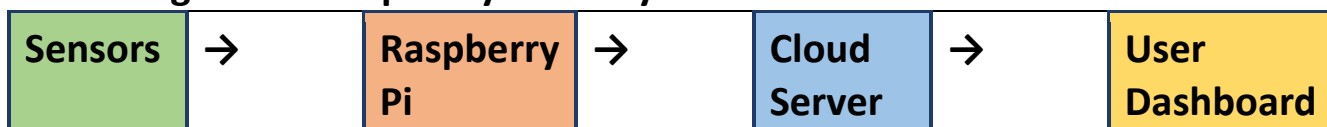
Step 7: View data on dashboard (ThingSpeak, Grafana).

Step 8: Add control logic (e.g., turn on fan if temperature is high).

Step 9: Add security (SSH keys, HTTPS, firewall).

Step 10: Test the complete system and deploy.

Block Diagram — Raspberry Pi IoT System:



Q7. Discuss the role of Linux in managing Raspberry Pi for IoT applications. What are the advantages of using Linux on Raspberry Pi? [4+4]

Role of Linux in Managing Raspberry Pi:

Raspberry Pi mostly uses Raspberry Pi OS (a Linux-based operating system). Linux plays a very important role in managing the hardware, software, and networking on Raspberry Pi.

1. Resource Management: Linux manages CPU, memory, storage, and I/O of Raspberry Pi efficiently.

2. Process Management: Linux runs multiple IoT programs at same time (multitasking).

3. Networking: Provides Wi-Fi, Ethernet, and Bluetooth support to connect with sensors and cloud.

4. Device Drivers: Linux has built-in drivers for GPIO, USB, camera, and sensors.

5. Security Management: Provides user permissions, firewall, and SSH for secure remote access.

Advantages of Using Linux on Raspberry Pi:

- **Free and Open-Source:** No license cost — anyone can use and modify.
- **Lightweight:** Runs smoothly on small Raspberry Pi hardware.
- **Stable and Reliable:** Does not crash easily; good for 24/7 IoT systems.
- **Large Community Support:** Millions of developers share help and code online.
- **Programming Support:** Supports Python, C, Java, Node.js — perfect for IoT.
- **Customizable:** Users can install only required software.
- **Security Features:** Built-in firewall, SSH, user permissions.

Q8. Discuss the various vulnerabilities of IoT devices and networks. How do these vulnerabilities impact the overall security of IoT system? [8]

Vulnerabilities of IoT Devices and Networks:

IoT vulnerabilities are weak points in devices, network, or software that hackers can exploit to attack the system.

- 1. Weak/Default Passwords:** Many IoT devices use factory default passwords like 'admin/admin' which are easy to guess.
- 2. Lack of Encryption:** Data sent in plain text can be intercepted by hackers.
- 3. Outdated Firmware:** Old firmware has known bugs that hackers exploit.
- 4. Insecure APIs:** Poorly designed APIs allow unauthorized access to device data.

Impact on Overall IoT Security:

- **Data Theft:** Personal and business data can be stolen.
- **Privacy Loss:** Cameras and mics can be hijacked to spy on users.
- **System Downtime:** DDoS attacks crash IoT services.
- **Financial Loss:** Hackers demand ransom or steal money via IoT.
- **Physical Danger:** Attacks on medical or vehicle IoT can harm human life.
- **Reputation Damage:** Companies lose customer trust after breaches..

Q9. Explain the design constraints and challenges faced in implementing IoT solutions in real world scenarios. Mention the challenges faced in countries like Nepal. [8]

Design Constraints & Challenges in IoT Implementation:

- 1. Power Consumption:** IoT devices need low power to run on batteries for long time.
- 2. Cost:** Sensors, boards, and cloud services increase overall cost.
- 3. Scalability:** System should handle millions of devices as it grows.
- 4. Connectivity:** Stable internet is needed which may not be available everywhere.
- 5. Interoperability:** Devices from different vendors may not work together.
- 6. Data Storage:** IoT generates huge data that needs big cloud storage.

Challenges Faced in Nepal:

- 1. Poor Internet Infrastructure:** Slow and unstable internet, especially in rural areas.
- 2. Load Shedding / Power Issues:** Frequent power cuts affect IoT device operation.
- 3. Geographical Difficulty:** Mountainous terrain makes network coverage difficult.
- 4. Low Awareness:** General public and businesses lack IoT knowledge.
- 5. High Cost of Devices:** IoT sensors and boards are expensive due to import.

Q10. Write short notes on: (any TWO) [2×4]

a) Smart Grid:

Smart Grid is a modern electricity network that uses IoT sensors, meters, and communication technology to monitor, control, and manage electricity flow efficiently between power stations and consumers.

Features:

Real-time electricity monitoring.
Automatic fault detection.
Smart meters at homes.
Integration with renewable energy (solar, wind).
Reduces power loss and improves supply.
Two-way communication between users and utility.

b) e-Healthy Body Area Network (BAN):

Body Area Network (BAN) is a wireless network of small IoT devices worn on or implanted in the human body to monitor health in real-time.

Devices:

Smart watches (heart rate, sleep monitor).
ECG patches.
Glucose monitors.
Blood pressure sensors.
Fall detectors for elderly.

Uses:

Remote patient monitoring.
Emergency alerts to doctors.
Fitness tracking.
Chronic disease management.

c) Amazon Web Services (AWS) for IoT:

AWS IoT is a cloud platform from Amazon that helps connect, manage, and secure billions of IoT devices and process their data.

Key AWS IoT Services:

AWS IoT Core: Connects IoT devices to cloud securely using MQTT/HTTP.

AWS IoT Analytics: Analyzes IoT data to find patterns.

AWS IoT Greengrass: Runs code on edge devices even offline.

AWS IoT Device Defender: Provides IoT security monitoring.

AWS IoT SiteWise: Manages industrial IoT data.

d) Smart Cities:

Smart Cities use IoT, sensors, and internet to improve city services like traffic, waste management, energy, water, and public safety.

Smart City Applications:

Smart Traffic: Automatic traffic lights, congestion control.

Smart Lighting: Street lights auto-adjust brightness.

Smart Parking: Sensors show empty parking slots.

Smart Waste: Bins tell when they are full.

Smart Water: Monitors water leakage and quality.

Public Safety: CCTV, emergency alarms with IoT.

Examples: Singapore, Barcelona, Dubai, Amsterdam.

— End of Notes —

All the Best for Your Exam! 📚 |

INTERNET OF THINGS (IoT)

Important Questions & Answers — Exam Notes

22 Important Questions | Simple Student Language | With Block Diagrams

Q.1: Define IoT. What are the characteristics of IoT. Explain with example.

IoT (Internet of Things) is a network of physical devices, vehicles, home appliances, and other objects connected to the internet, which can collect and exchange data with each other without human intervention

Characteristics of IoT

- 1. Connectivity:** All IoT devices are connected to internet or network.
- 2. Intelligence:** IoT devices can take smart decisions using sensors and AI.
- 3. Sensing:** Devices use sensors to collect data from environment (temperature, motion, etc.).
- 4. Scalability:** Number of connected devices can be increased easily.
- 5. Dynamic Nature:** Devices can change state — on/off, active/sleep — automatically.
- 6. Heterogeneity:** Different types of devices with different hardware work together.
- 7. Security:** Data protection and privacy are important features.
- 8. Self-Configuring:** Devices can setup themselves automatically.

Example

Smart Home Example: In a smart home, the AC, lights, fan, and door lock are connected to the internet. When you leave the house, your phone sends a signal — the AC turns off, lights switch off, and door locks automatically. All this happens without you touching anything. This is IoT in real life.

Q.2: What are the benefits of IoT device.

Benefits of IoT Devices

- 1. Automation:** IoT automates daily tasks like switching lights, controlling AC, watering plants.
- 2. Time Saving:** Reduces manual work and saves user's time.
- 3. Cost Saving:** Reduces electricity, fuel, and labor costs.
- 4. Better Decision Making:** Real-time data helps in taking quick and correct decisions.
- 5. Improved Efficiency:** Machines work more accurately and productively.
- 6. Remote Control:** Devices can be controlled from anywhere using mobile.
- 7. Improved Safety:** Smart alarms, cameras, and sensors improve security.
- 8. Healthcare Support:** Smart watches monitor heart rate, BP, and alert doctors.
- 9. Better Customer Experience:** Personalized services in shops, hotels, transport.
- 10. Environment Friendly:** Reduces power waste and pollution through smart control.

PUBLISHER

Q.3: Explain the process of identification of IoT objects and services and its importance in IoT ecosystems.

Identification of IoT Objects and Services

Identification in IoT means giving each device a unique name or address so it can be recognized in the network. Every IoT object (like sensor, camera, fridge) needs a unique ID so that data can be sent and received correctly.

Process of Identification

- 1. Naming:** Every device is given a unique name (like device_01, sensor_A).
- 2. Addressing:** Each device gets a unique IP address (IPv4 or IPv6) for communication.
- 3. RFID / Barcode:** Physical objects are identified using RFID tags or barcodes.
- 4. MAC Address:** Hardware identity of each device is stored using MAC address.
- 5. UUID (Universal Unique ID):** Global unique ID assigned to each object.
- 6. Discovery Service:** Network finds and lists all connected devices automatically.

Importance in IoT Ecosystem

- Helps devices to communicate without confusion.
- Ensures correct data reaches the correct device.
- Makes tracking and monitoring easy.
- Improves security by identifying authorized devices.
- Supports scalability — millions of devices can join network.

Helps in managing and controlling devices remotely.

Q.4: Describe the role of security and privacy in IoT. Why is it challenging to maintain these in IoT systems?

Role of Security in IoT

Security in IoT means protecting devices, data, and networks from hackers, viruses, and unauthorized access. Since IoT devices handle personal information (like health, location, home camera), strong security is very important.

Role of Privacy in IoT

Privacy in IoT means keeping user's personal data safe and not sharing it without permission. IoT devices collect lots of personal data — where you go, what you buy, when you sleep. This data must be kept private.

Why is it Challenging?

- 1. Large Number of Devices:** Billions of devices are connected — hard to secure each one.
- 2. Limited Resources:** Small IoT devices have less memory and CPU, so strong encryption is difficult.

3. Different Standards: Many companies make devices with different rules — no common standard.

4. Old Devices: Many IoT devices don't get software updates, becoming easy targets.

5. Weak Passwords: Users often use default or weak passwords.

6. Data Leakage: Data travels through many networks — easy to intercept.

7. Physical Attack: Devices are placed in open areas — can be stolen or damaged.

8. Privacy Laws: Different countries have different privacy rules.

Q.5: Describe device intelligence and communication capabilities in IoT.

Device Intelligence in IoT

Device Intelligence means the ability of IoT devices to think, learn, and take decisions on their own using data. Smart devices use AI, machine learning, and sensors to become intelligent.

Features of Device Intelligence

- Data Collection using sensors.
- Data Processing using microcontrollers.
- Learning from past data (Machine Learning).
- Automatic decision making.
- Predicting future events (like rain, machine failure).
- Adapting to environment changes.

Communication Capabilities in IoT

Communication capability means how IoT devices talk to each other and to servers. They use different wireless and wired technologies for communication.

1. Wi-Fi: For high-speed short-range communication in homes and offices.

2. Bluetooth: For very short-range device-to-device connection (e.g., earbuds).

3. Zigbee: Low-power short-range communication for smart home devices.

4. 4G/5G Mobile Network: For long-distance mobile IoT devices.

5. LoRa / LPWAN: For long distance, low power communication.

6. NFC (Near Field Communication): For very close-range payments and pairing.

7. RFID: For object identification without contact.

8. Satellite: For remote areas where mobile network is not available.

Q.6: Analyze the key environmental and traffic characteristics in IoT and their impact on IoT applications.

Environmental Characteristics in IoT

- 1. Temperature:** Devices must work in hot and cold environments (e.g., outdoor sensors).
- 2. Humidity:** Moisture level affects sensor accuracy.
- 3. Location:** Indoor or outdoor placement affects signal strength.
- 4. Power Supply:** Some places don't have electricity — devices use battery/solar.
- 5. Physical Damage Risk:** Devices in open areas may face rain, dust, or theft.
- 6. Network Availability:** Rural areas may have weak internet signal.

Traffic Characteristics in IoT

- 1. Small Data Packets:** IoT devices usually send small amounts of data (like temperature reading).
- 2. High Frequency:** Data is sent very frequently (every few seconds).
- 3. Real-Time:** Data must reach quickly for time-sensitive applications.
- 4. Uplink Heavy:** More data goes from device to cloud than cloud to device.
- 5. Burst Traffic:** Sometimes many devices send data at the same time.
- 6. Reliability:** Data loss must be avoided in critical systems (health, safety).

Impact on IoT Applications

These characteristics affect the design of IoT systems. For example, in smart agriculture, sensors must withstand rain and heat (environmental), and send data every few minutes (traffic). Poor environment = weak signal = wrong data = wrong decision. So proper design is important.

Q.7: What do you mean by Environment characteristics of IoT? Explain it.

Meaning

Environment characteristics of IoT refer to the physical conditions in which IoT devices are deployed and how these conditions affect their working. IoT devices operate in different environments — homes, factories, farms, hospitals, roads — each with different conditions like temperature, humidity, dust, signal strength, and power.

Main Environment Characteristics

- 1. Temperature Variation:** Devices must work in hot summer or cold winter without failing.
- 2. Humidity & Moisture:** Water and moisture can damage sensitive electronics.
- 3. Dust and Dirt:** Devices in factories or roads need dust-proof design.
- 4. Location (Indoor/Outdoor):** Outdoor devices face weather; indoor devices face limited signal.

Q.8: Define open architecture. Explain with neat diagram.

Open Architecture in IoT is a system design where the standards, interfaces, and protocols are open (publicly available) so that devices from different companies can work together easily. It follows common standards like MQTT, HTTP, and CoAP — allowing any manufacturer to build compatible IoT devices.

Features of Open Architecture

- Uses open standards and protocols.
- Interoperable — devices from different vendors work together.
- Scalable — easy to add new devices.
- Flexible — supports different applications.
- Vendor-neutral — no dependency on one company.
- Cost-effective and easy to modify.

Diagram of IoT Open Architecture (4 Layers)

Internet of Things (IoT) Architecture

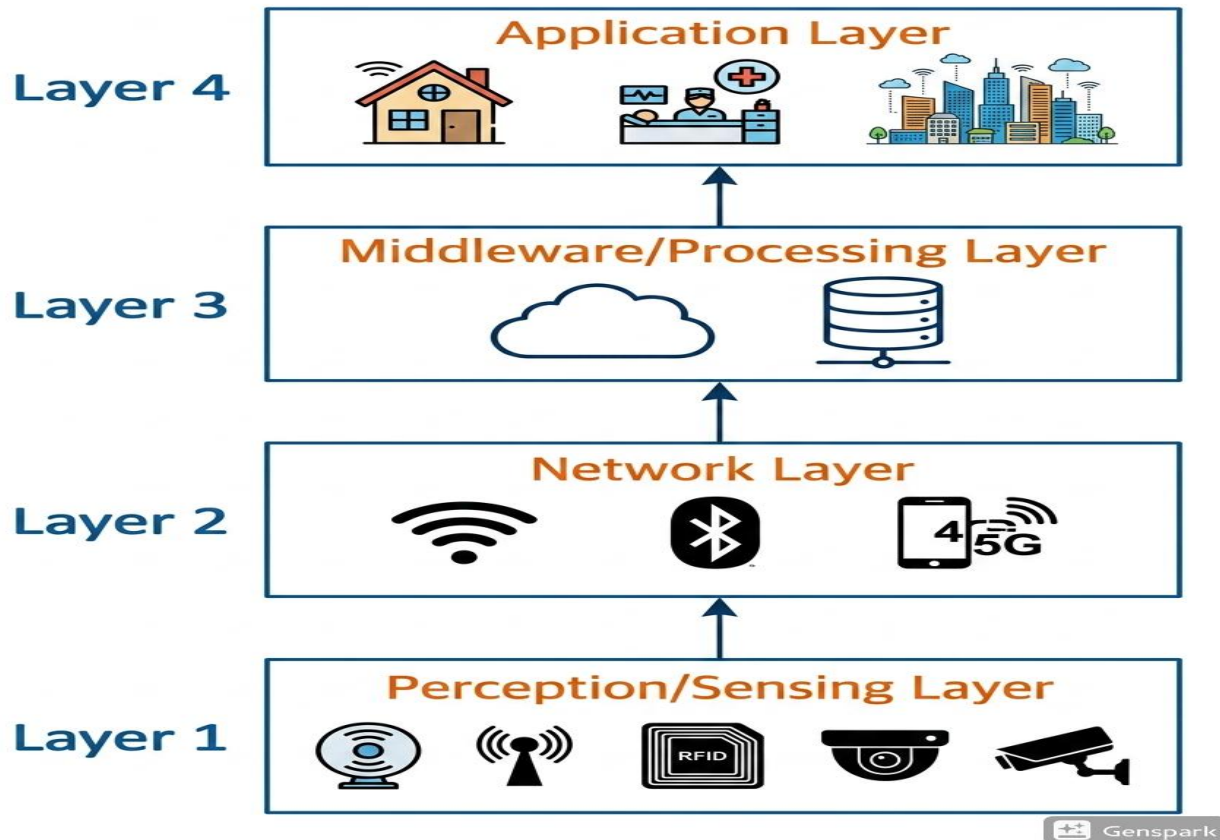


Fig: IoT Open Architecture — 4 Layer Model

Explanation of Layers

- 1. Perception Layer (Bottom):** Physical devices, sensors, RFID, cameras — collect data from environment.
- 2. Network Layer:** Transfers data using Wi-Fi, Bluetooth, 4G/5G, Zigbee.
- 3. Middleware / Processing Layer:** Processes and stores data using cloud and databases.
- 4. Application Layer (Top):** Provides services to users — smart home, healthcare, smart city apps.

Q.9: What do you mean by RFID technology? Explain it.

RFID (Radio Frequency Identification) is a wireless technology that uses radio waves to automatically identify and track objects using tags and readers. It works without physical contact — the reader can identify an object from a distance.

Components of RFID System

1. **RFID Tag:** Small chip attached to the object, stores unique ID.
2. **RFID Reader:** Device that sends radio waves and reads the tag.
3. **Antenna:** Sends and receives radio signals between tag and reader.
4. **Host Computer / Database:** Stores the information read from tags.

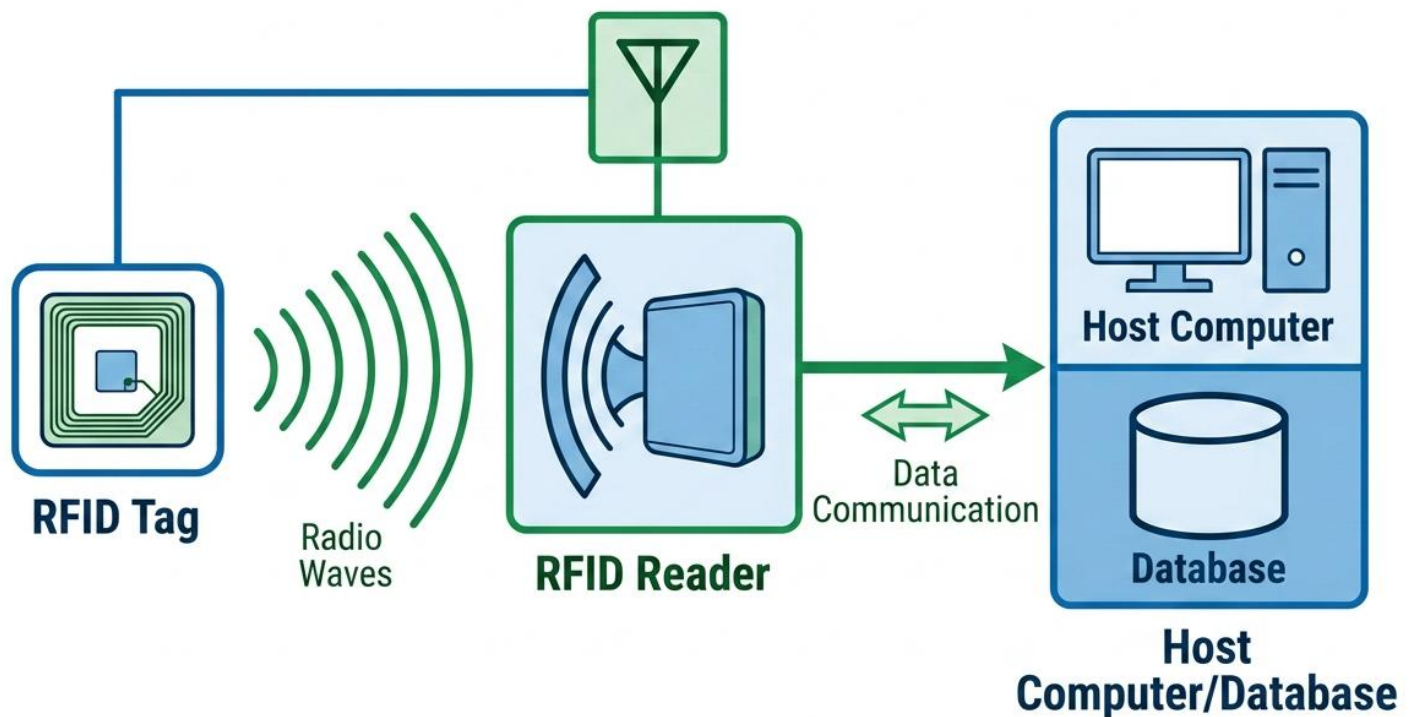


Fig: Working of RFID Technology

Working of RFID

1) Reader sends radio waves through antenna. 2) When RFID tag comes near, it receives the signal and sends back its unique ID. 3) Reader receives the ID and sends it to the computer. 4) Computer matches the ID with database and gives the object's information.

Uses of RFID

Used in shopping malls (barcode replacement), toll systems (Fastag), library book tracking, employee ID cards, animal tracking, and inventory management.

Q.10: Explain RFID and satellite technology as used in IoT. Discuss their use cases and limitations.

RFID in IoT

RFID uses radio waves to identify and track objects. In IoT, RFID tags are placed on physical things so they can be identified and monitored automatically.

Use Cases of RFID

- Smart shopping — auto billing (Amazon Go store).
- Toll collection on highways (Fastag).
- Warehouse inventory tracking.
- Library book management.
- Animal tracking in farms.

Limitations of RFID

- Short range — only few meters.
- Signal blocked by metal and water.
- Security risk — tags can be cloned.
- Cost is high for large-scale use.

Satellite Technology in IoT

Satellite technology uses satellites in space to send and receive data over very long distances. It is used in IoT for remote areas where mobile or Wi-Fi signal is not available.

- **Use Cases of Satellite**
- GPS navigation (Google Maps, vehicle tracking).
- Agriculture monitoring in remote farms.
- Weather forecasting.
- Disaster management and rescue operations.
- Ship and aircraft tracking.
- Military and defense applications.

Limitations of Satellite

- Very high cost of satellite and equipment.
- High data delay (latency).
- Weather affects signal quality.
- Limited bandwidth compared to fiber optics.

Q.11: Explain M2M (Machine-to-Machine) and WSN (Wireless Sensor Network) protocols.

M2M (Machine-to-Machine)

M2M means direct communication between two or more machines without human involvement. Machines exchange data using wired or wireless networks to perform tasks automatically.

Features of M2M

- Direct machine-to-machine communication.
- No human interaction needed.
- Uses cellular, Wi-Fi, or wired networks.
- Real-time data exchange.
- Automatic decisions and actions.

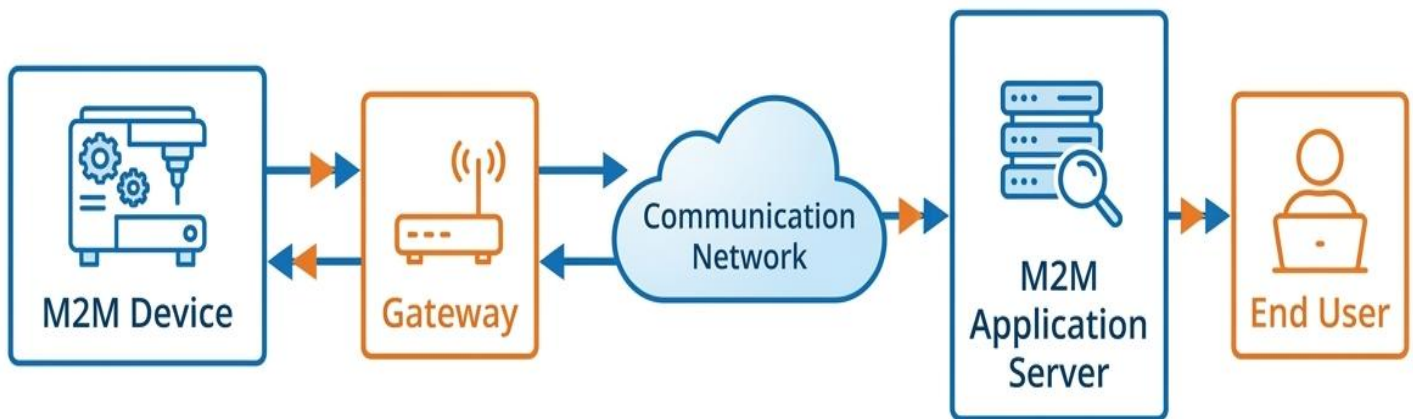


Fig: M2M (Machine-to-Machine) Communication

WSN (Wireless Sensor Network)

WSN is a group of small sensor nodes spread over an area that collect data (like temperature, humidity, motion) and send it wirelessly to a central computer for analysis.

Features of WSN

- Made up of many small wireless sensor nodes.
- Low power consumption (battery operated).
- Nodes communicate with each other in mesh network.
- Used to monitor physical conditions.
- Self-organizing and self-healing network.

Difference between M2M and WSN

S.N	Basis	M2M	WSN
.			

1	Meaning	Direct communication between two machines.	Network of many sensors sending data wirelessly.
2	Nodes	Uses machines/devices.	Uses small sensor nodes.
3	Power	High power devices.	Very low power (battery).
4	Range	Can be long-range.	Usually short-range.
5	Example	Vending machine to server.	Smart farm with soil sensors.

Q.12: What do you mean by Protocols in IoT, explain any.

Meaning of Protocol in IoT

Protocol is a set of rules that decides how data will be sent, received, and understood between two IoT devices. Without protocol, devices cannot communicate properly.

Important IoT Protocols

- 1. MQTT (Message Queuing Telemetry Transport):** Lightweight publish-subscribe protocol; best for low-power devices; used in smart home.
- 2. CoAP (Constrained Application Protocol):** Web-based protocol for small devices with limited memory and battery.
- 3. HTTP / HTTPS:** Standard web protocol; used when devices need to talk to web servers.
- 4. AMQP (Advanced Message Queuing Protocol):** Reliable protocol for sending messages between systems.
- 5. Zigbee:** Wireless protocol for short-range low-power communication.
- 6. Bluetooth / BLE:** Short-range wireless protocol; BLE is battery-friendly version.
- 7. 6LoWPAN:** Allows small IoT devices to use IPv6 addressing.
- 8. LoRaWAN:** Long-range low-power protocol used in smart cities.

Explanation of MQTT (Example)

MQTT works on publish-subscribe model. A device (publisher) sends data to a broker (server), and other devices (subscribers) receive that data from the broker. It is very lightweight, uses less battery, and works well on slow networks. That's why it is popular in IoT applications like smart home and industrial IoT.

Q.13: What do you mean by Pi? Is there any difference between general purpose computer and Pi (Raspberry Pi)?

Definition of Raspberry Pi (Pi)

Raspberry Pi is a small, low-cost, credit-card sized single-board computer developed by Raspberry Pi Foundation (UK). It can perform many tasks like a normal computer — running programs, connecting to internet, and controlling hardware. It is widely used in IoT projects, robotics, and learning programming.

Difference between General Purpose Computer and Raspberry Pi

S.N.	Basis	General Purpose Computer	Raspberry Pi
1	Size	Large (desktop/laptop).	Very small — credit card size.
2	Cost	Expensive (Rs. 40,000+).	Cheap (Rs. 5,000 - 12,000).
3	Power	Uses lot of electricity.	Very low power (5V).
4	Processor	Powerful Intel/AMD CPU.	ARM-based low-power CPU.
5	RAM	8 GB to 32 GB.	512 MB to 8 GB only.
6	Storage	Hard disk / SSD.	MicroSD card.
7	OS	Windows, macOS, Linux.	Raspberry Pi OS (Linux-based).
8	GPIO Pins	Not available.	40 GPIO pins for hardware control.
9	Use	Office, gaming, heavy tasks.	IoT, robotics, learning.
10	Portability	Less portable.	Highly portable and small.

Q.14: Explain the IoT architecture and its components in detail.

IoT Architecture

IoT Architecture is a layered structure that shows how IoT devices, networks, and applications work together. The most common model has 4 layers.

Internet of Things (IoT) Architecture

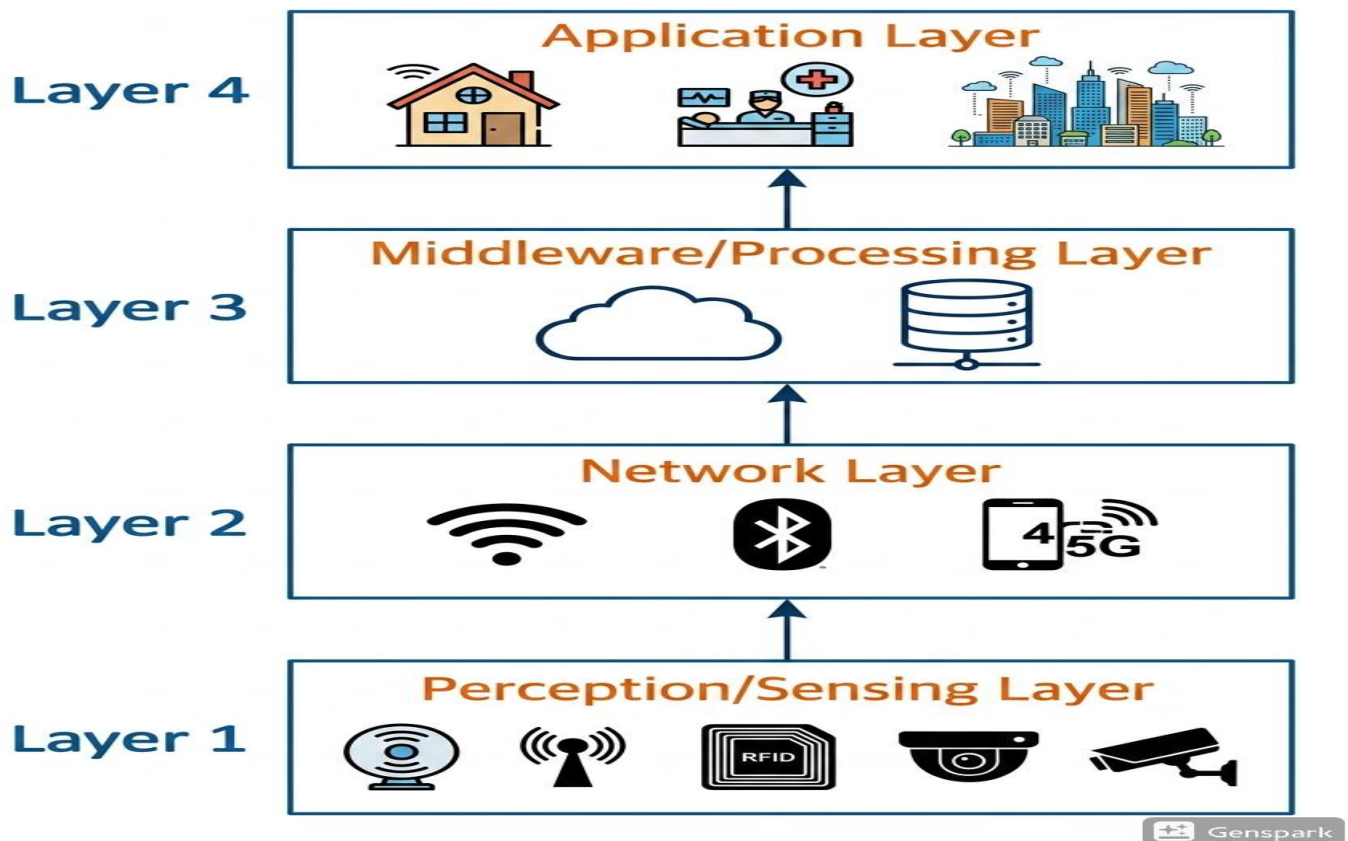


Fig: 4-Layer IoT Architecture

1. Perception Layer (Sensing Layer)

This is the bottom layer that collects data from the physical world. It contains sensors, actuators, RFID tags, GPS, and cameras. Example: temperature sensor in AC.

2. Network Layer (Transport Layer)

This layer transfers data collected by sensors to the processing layer using different networks like Wi-Fi, Bluetooth, 4G/5G, Zigbee, and LoRa.

3. Middleware Layer (Processing Layer)

This layer processes, stores, and analyzes the data using cloud computing, big data, and databases. It also uses AI and Machine Learning to make smart decisions.

4. Application Layer

This is the top layer that provides services to users through applications. Examples: smart home apps, healthcare monitoring, smart agriculture, smart city, and industrial automation.

Main Components of IoT

- 1. Sensors:** Collect real-world data (temperature, motion, light).
- 2. Actuators:** Take physical action based on data (turn on fan, open door).
- 3. Gateway:** Connects local devices to internet.
- 4. Cloud:** Stores and processes data on remote servers.
- 5. User Interface:** Mobile app or website to control and monitor devices.
- 6. Network:** Wi-Fi, mobile, or wired connection.
- 7. Protocols:** Rules for communication (MQTT, HTTP, CoAP).

Q.15: Describe the various IoT frameworks and their significance in building IoT applications.

What is IoT Framework?

An IoT Framework is a ready-made platform that provides tools, libraries, and services to easily build, connect, and manage IoT devices and applications. It saves developers' time and effort.

Popular IoT Frameworks

- 1. AWS IoT (Amazon):** Cloud-based platform to connect and manage millions of IoT devices; supports MQTT and HTTPS.
- 2. Microsoft Azure IoT:** Microsoft's cloud platform for building smart IoT applications with AI support.
- 3. Google Cloud IoT:** Google's platform with strong data analytics and machine learning.
- 4. IBM Watson IoT:** IBM's AI-powered IoT platform for enterprise applications.

Significance of IoT Frameworks

- Fast Development — reduces coding time.
- Ready-made Security — built-in encryption and authentication.
- Scalability — can handle millions of devices.
- Easy Integration with cloud, AI, and big data.
- Remote Device Management from anywhere.
- Data Analytics tools to understand data.
- Cost-effective — no need to build from scratch.

Q.16: What are the main benefits of IoT in modern applications? Provide examples to illustrate these benefits.

Main Benefits of IoT

- 1. Automation:** Devices work automatically without human help. Example: Smart AC turns on when you enter home.
- 2. Real-Time Monitoring:** Data is available live 24×7. Example: Doctors monitor patient's heart rate from hospital.
- 3. Better Decision Making:** Data helps to take smart decisions. Example: Traffic system controls signal based on live traffic.
- 4. Cost Saving:** Reduces electricity, fuel, and manpower cost. Example: Smart street lights turn off when no one is around.
- 5. Improved Safety:** Alerts users about danger. Example: Smoke sensor sends alert if fire starts.

- 6. Increased Productivity:** Machines work faster and more accurately. Example: Smart factory produces more with less labor.
- 7. Better Healthcare:** Continuous health monitoring saves lives. Example: Smart watch tracks BP and heart rate.
- 8. Environment Protection:** Reduces power waste and pollution. Example: Smart irrigation waters plants only when needed.
- 9. Convenience:** Makes life easy and comfortable. Example: Voice-controlled home with Alexa.
- 10. Remote Control:** Control devices from anywhere. Example: Turn off geyser using phone.



Q.17: Analyze the Zigbee architecture and its application in IoT.

What is Zigbee?

Zigbee is a low-power, low-cost wireless communication technology based on IEEE 802.15.4 standard, mainly used for short-range communication between IoT devices in smart homes and industries.

Zigbee Architecture (Components)

1. **Zigbee Coordinator (ZC):** The main device that controls the whole network. Only ONE coordinator per network.
2. **Zigbee Router (ZR):** Forwards data between devices; extends network range.
3. **Zigbee End Device (ZED):** Simple low-power device that only sends/receives data (like a light bulb, sensor).

Zigbee Network Architecture (Mesh Topology)

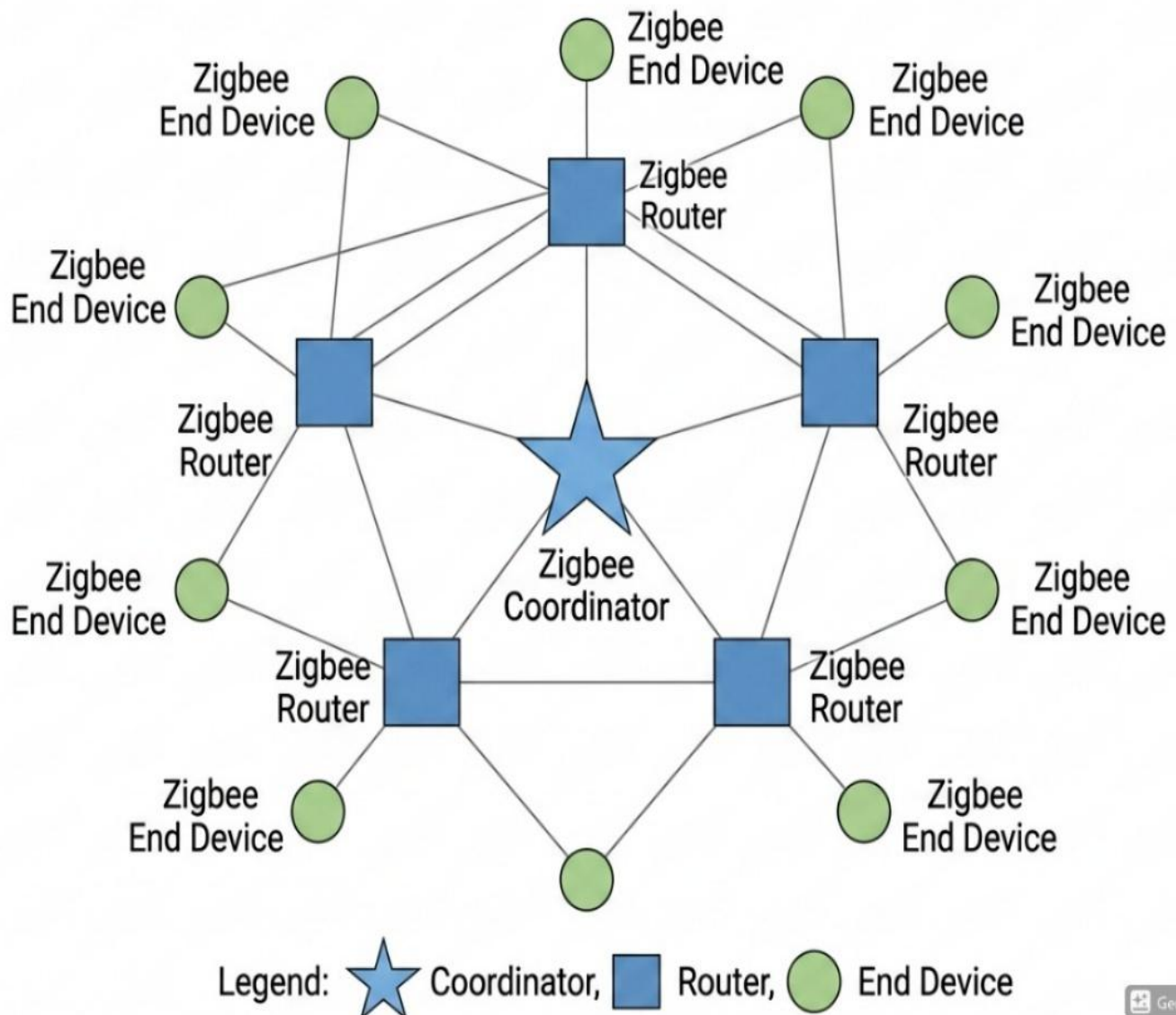


Fig: Zigbee Mesh Network Architecture

Features of Zigbee

- Low power consumption — battery lasts for years.
- Short range — 10 to 100 meters.
- Low data rate — 250 Kbps.
- Supports mesh, star, and tree topologies.
- Can connect up to 65,000 devices in one network.
- Secure with AES-128 encryption.

Applications in IoT

- Smart home — lights, fans, locks, thermostats.
- Smart lighting systems (Philips Hue).
- Industrial automation — machine monitoring.
- Healthcare — patient monitoring devices.
- Smart agriculture — soil sensors.
- Smart metering — electricity, water meters.

Q.18: Describe the CoAP (Constrained Application Protocol) and its importance in IoT communication.

Definition of CoAP

CoAP (Constrained Application Protocol) is a special web transfer protocol designed for small IoT devices that have limited memory, low power, and slow networks. It works similarly to HTTP but is much lighter and faster.

Features of CoAP

- Lightweight — small message size, saves bandwidth.
- Uses UDP (not TCP) — faster and less power hungry.
- Request/Response model like HTTP (GET, POST, PUT, DELETE).
- Supports multicast — one message to many devices.
- Built-in security using DTLS (Datagram TLS).
- Low latency — quick response time.
- Designed for machine-to-machine communication.

Importance of CoAP in IoT

- Perfect for small battery-powered devices.
- Works well in unreliable networks.
- Uses less data — saves internet cost.
- Compatible with HTTP for easy web integration.
- Enables real-time communication.
- Reduces device processing load.

Q.19: Discuss the role of IEEE 802.15.4 in IoT networking. What are its main features and limitations?

Role of IEEE 802.15.4 in IoT

IEEE 802.15.4 is a wireless communication standard designed for low-rate, low-power, and low-cost devices. It is the base of many popular IoT technologies like Zigbee, 6LoWPAN, and Thread. It defines how small devices communicate wirelessly over short distances.

Main Features of IEEE 802.15.4

- 1. Low Power:** Devices consume very little battery.
- 2. Low Data Rate:** Up to 250 Kbps — enough for sensors.
- 3. Short Range:** Usually 10 to 100 meters.
- 4. Frequency Bands:** Uses 2.4 GHz, 915 MHz, and 868 MHz.
- 5. Mesh Topology:** Supports star, peer-to-peer, and mesh networks.
- 6. Low Cost:** Cheap hardware and easy to deploy.
- 7. Security:** Uses AES-128 encryption for data protection.
- 8. Reliability:** CSMA/CA method avoids data collision.

Limitations of IEEE 802.15.4

- Low data rate — cannot handle video or audio streaming.
- Short range — not good for long-distance communication.
- Interference — 2.4 GHz band is shared with Wi-Fi and Bluetooth.
- Small packet size (127 bytes) — limits large data transfer.
- Weak signal through walls and metal.
- Battery limitation — sleep mode needed for long life.

Q.20: Explain how Raspberry Pi can be used to build an IoT system. Include examples of components.

Raspberry Pi in IoT

Raspberry Pi is a small computer that acts as the brain of an IoT system. It reads data from sensors, processes it, connects to internet, and controls other devices. Because of its low cost, small size, and GPIO pins, it is very popular for building IoT projects.

Raspberry Pi Board Layout: Educational Block Diagram

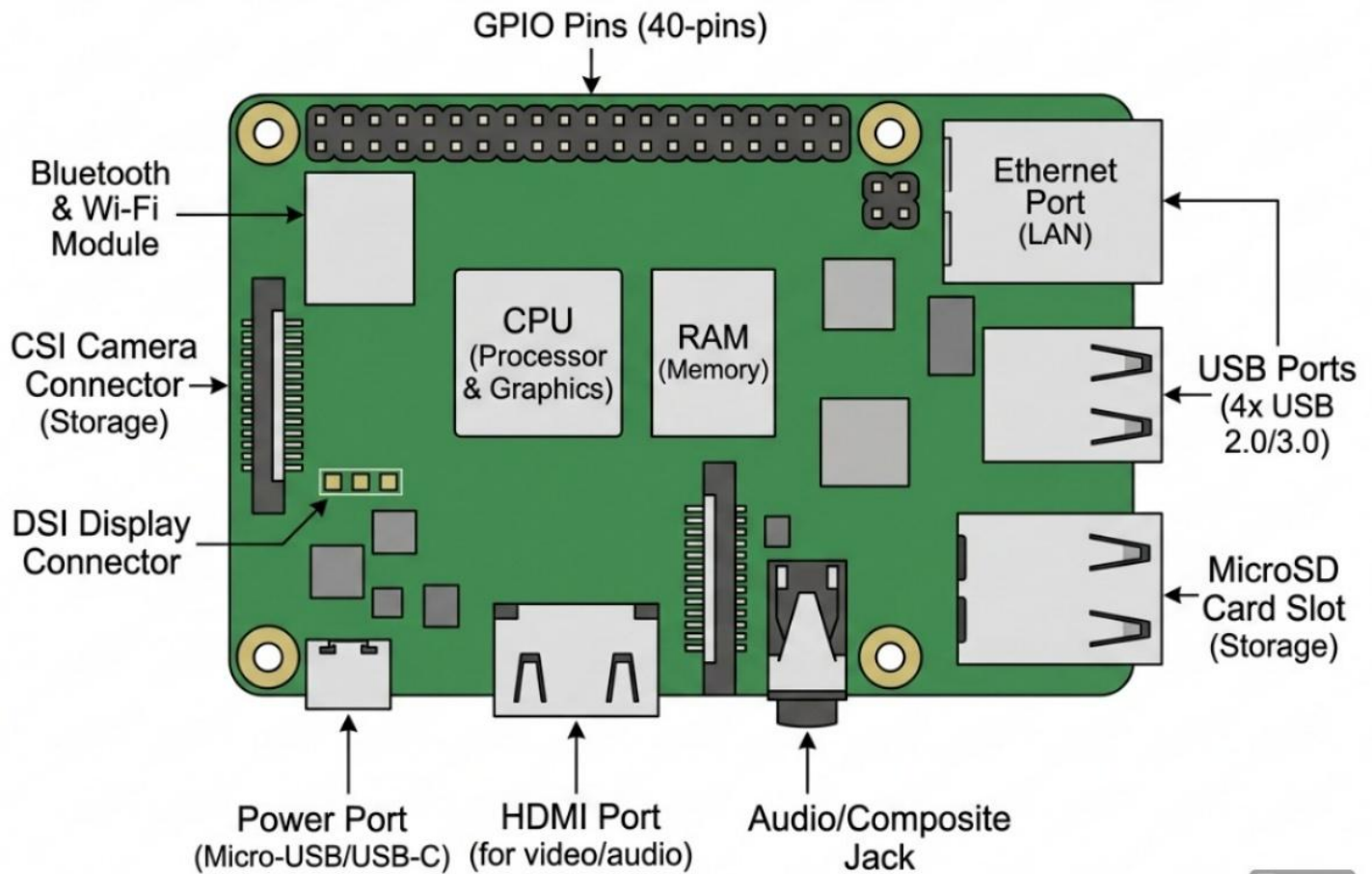


Fig: Raspberry Pi Board and its Components

Components Used with Raspberry Pi in IoT

- 1. Sensors:** DHT11 (temperature/humidity), PIR (motion), LDR (light), Ultrasonic (distance).
- 2. Actuators:** LED bulbs, DC motors, servo motors, relay switches, buzzers.
- 3. Communication Modules:** Wi-Fi module (built-in), Bluetooth, Zigbee, GSM (for mobile network).
- 4. Camera Module:** Pi Camera for image/video capture in surveillance.
- 5. Display:** LCD screen or touchscreen to show data.
- 6. Power Supply:** 5V USB adapter or battery.
- 7. Storage:** MicroSD card for OS and data storage.
- 8. Cloud Service:** AWS IoT, ThingSpeak, Firebase for data storage online.

Example: Smart Home Automation using Raspberry Pi

Steps: (1) Connect temperature sensor and PIR motion sensor to Pi's GPIO pins. (2) Write Python program to read sensor data. (3) Send data to cloud (ThingSpeak) using Wi-Fi. (4) Create a mobile app to view data and control fan/light. (5) Pi automatically turns ON fan if temperature is high or lights if motion is detected.

Q.21: Discuss the significance of Raspberry Pi interfaces for IoT applications. Give examples of common interfaces used.

What are Raspberry Pi Interfaces?

Interfaces are the ports and pins on Raspberry Pi that allow it to connect and communicate with sensors, actuators, and other devices. These interfaces make Raspberry Pi very useful for IoT projects.

Common Interfaces of Raspberry Pi

- 1. GPIO (General Purpose Input/Output):** 40 pins used to connect sensors, LEDs, motors. Can be used as input or output.
- 2. USB Ports:** Connect keyboard, mouse, USB camera, or Wi-Fi/Bluetooth dongles.
- 3. HDMI Port:** Connect monitor or TV to see graphical output.
- 4. Ethernet Port:** For wired internet connection.
- 5. Wi-Fi and Bluetooth (Built-in):** For wireless communication with IoT devices and internet.

Significance of Interfaces in IoT

- GPIO makes it easy to connect any sensor or actuator.
- Wi-Fi and Bluetooth enable wireless IoT communication.
- I2C and SPI allow multiple sensor connections.
- USB expands functionality with plug-and-play devices.
- Camera interface enables IoT vision projects.
- Ethernet ensures stable internet for critical IoT systems.

Q.22: Write a short notes on these (any Two)

(a) 6LoWPAN

6LoWPAN stands for 'IPv6 over Low-Power Wireless Personal Area Networks'. It is a protocol that allows small, low-power IoT devices to use IPv6 addressing over IEEE 802.15.4 wireless networks.

Key Points:

Enables tiny IoT devices to connect to the internet using IPv6.

Uses very low power — good for battery devices.

Supports huge number of devices (IPv6 has almost unlimited addresses).

Data compression reduces packet size for slow networks.

Used in smart home, smart lighting, industrial IoT.

Works over Zigbee and Thread networks.

(b) RPi GPIO (Raspberry Pi General Purpose Input/Output)

GPIO stands for General Purpose Input/Output. Raspberry Pi has 40 GPIO pins that can be programmed as input (to read data from sensors) or output (to control devices like LEDs, motors, buzzers).

Key Points:

Total 40 pins — some are power (3.3V, 5V), some are ground, rest are GPIO.

Can be used as INPUT (from sensors) or OUTPUT (to actuators).

Voltage: works on 3.3V — do NOT connect 5V directly (may damage).

Supports I2C, SPI, UART communication.

Programmed using Python (RPi.GPIO library) or C.

Used to connect sensors like temperature, motion, and to control LEDs, motors.

(c) Sensor Technology

Sensors are electronic devices that detect changes in the physical environment (like temperature, light, motion, pressure) and convert them into electrical signals.

They are the eyes and ears of IoT systems.

Types of Sensors:

Temperature Sensor — measures heat (e.g., DHT11, LM35).

Humidity Sensor — measures moisture in air.

Motion Sensor (PIR) — detects human movement.

Light Sensor (LDR) — detects light intensity.

Pressure Sensor — measures pressure of gas or liquid.

Ultrasonic Sensor — measures distance.

Gas Sensor — detects harmful gases (MQ series).

Proximity Sensor — detects nearby objects.

Accelerometer / Gyroscope — detects motion and rotation.

Camera Sensor — captures images and videos.

Applications:

Used in smart homes, healthcare devices, cars, mobile phones, weather stations, agriculture, industries, and security systems.

Q1. Define the Internet of Things (IoT) and discuss the benefits of IoT and provide examples of its applications in different industries.

The Internet of Things (IoT) is a network of physical devices embedded with sensors, software, and internet connectivity that collect and exchange data automatically to perform smart actions.

Benefits of IoT:

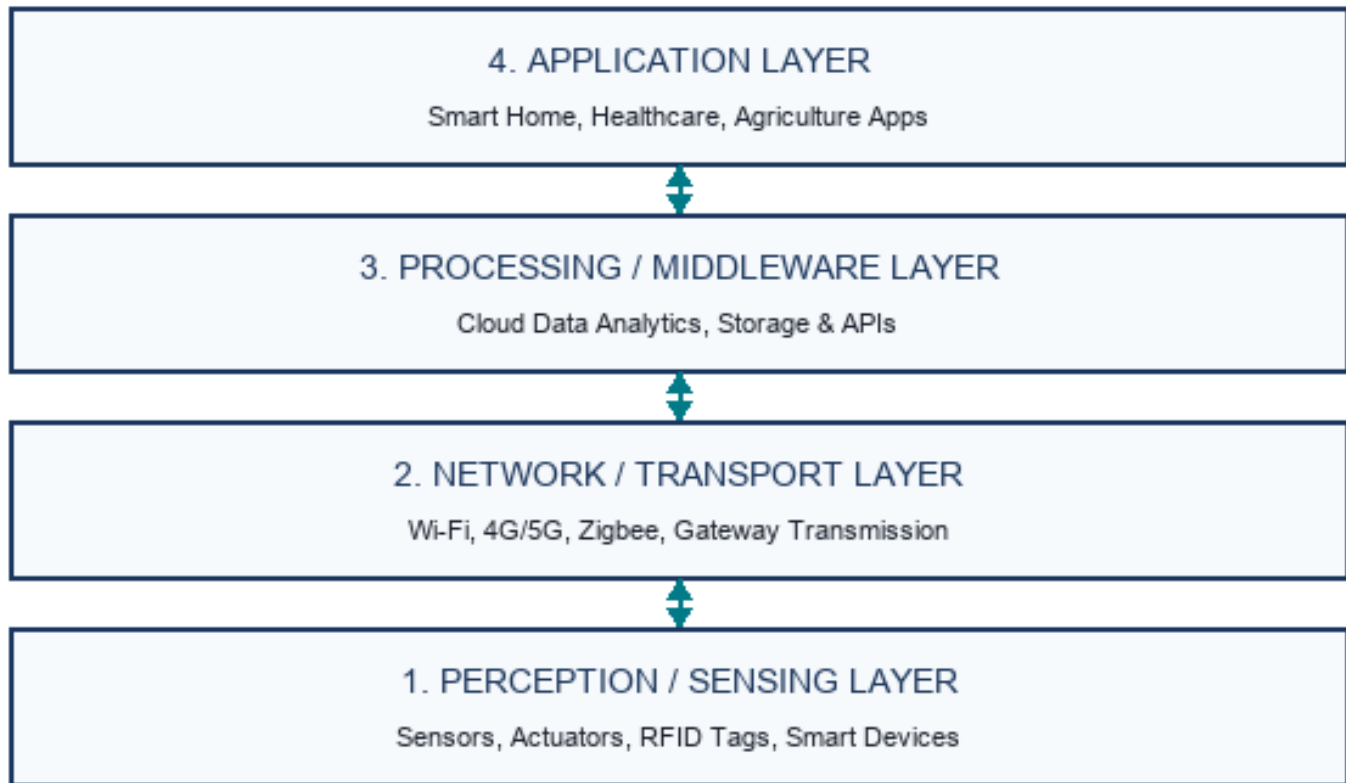
1. Automation: Machines work automatically without human manual operation.
2. Remote Monitoring: Monitor devices from anywhere using smartphone apps.
3. Cost & Energy Savings: Smart electricity meters reduce power wastage.
4. Improved Safety: Gas leak sensors alert users instantly before accidents.
5. Better Decision Making: Real-time data analytics helps businesses improve processes.

Applications in Different Industries:

Industry	IoT Application	Example
Agriculture	Smart Irrigation System	Soil moisture sensor waters crops automatically
Healthcare	Wearable Health Monitor	ECG band sends heart data to doctor remotely
Manufacturing	Predictive Maintenance	Vibration sensor detects machine failure early
Transport	Fleet GPS Tracking	Real-time truck location via 4G IoT tracker
Smart Home	Home Automation	Voice-controlled smart lights and AC systems

Q2. Explain the key components of the IoT architecture AND various IoT frameworks available.

Part A — IoT Architecture (Textbook Block Diagram):



4-LAYER IOT ARCHITECTURE BLOCK DIAGRAM

Fig: Textbook Block Diagram of 4-Layer IoT Architecture

Key Components of IoT Architecture:

1. Perception/Sensing Layer: Physical sensors (temperature, motion, gas) collect real-world data.
2. Network/Transport Layer: Transmits data via Wi-Fi, 4G/5G, Zigbee, Bluetooth to gateway.
3. Processing/Middleware Layer: Cloud servers store, clean, and analyze sensor data streams.
4. Application Layer: Delivers smart dashboards, alerts, and controls to end users.

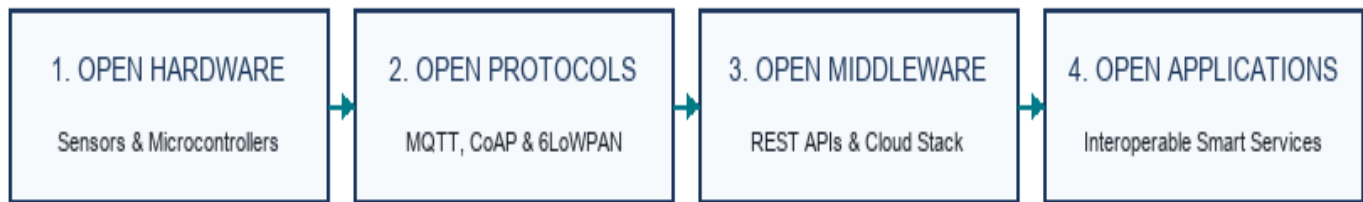
Various IoT Frameworks

An IoT framework is a set of ready-made tools, protocols, and software that helps developers build IoT applications quickly. Some popular IoT frameworks are:

- AWS IoT Core – Amazon's cloud based IoT platform for connecting and managing devices.
- Microsoft Azure IoT Suite – Provides cloud services for device management, analytics, and security.
- Google Cloud IoT – Used for connecting, processing, and analysing IoT data on Google Cloud.
- IBM Watson IoT Platform – Offers device management along with AI based data analytics.
- Node-RED – A flow based, open-source tool used to wire together IoT devices and APIs easily.
- Eclipse IoT – A group of open-source projects/frameworks for building IoT systems.

Q3. Describe Open Architecture AND Communication Capabilities of IoT systems.

— Open Architecture:



OPEN IOT ARCHITECTURE BLOCK DIAGRAM

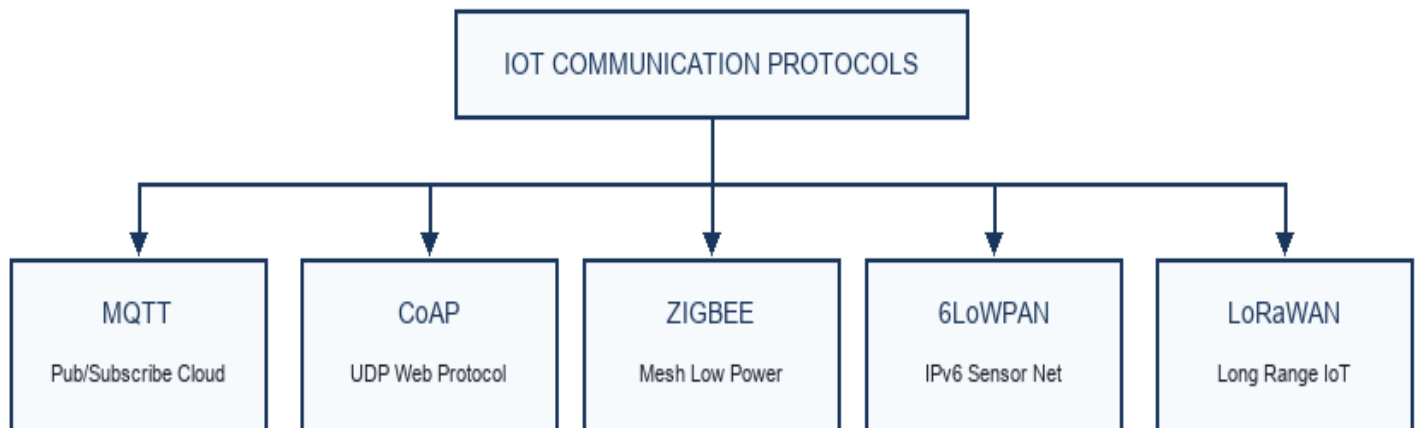
Fig: Open IoT Architecture Block Diagram

Open Architecture is a modular, vendor-neutral IoT design framework based on open standards and public APIs, allowing hardware and software from any manufacturer to integrate freely.

Key Features:

1. No vendor lock-in — mix any brands freely.
2. Open standards: MQTT, REST API, JSON, IPv6.
3. Easy to expand and upgrade over time.

Communication Capabilities:



MAJOR IOT COMMUNICATION PROTOCOLS OVERVIEW

Fig: IoT Communication Protocols Overview

1. Short Range: Bluetooth (BLE), Zigbee, Wi-Fi for smart home room-level control.
2. Medium Range: Z-Wave, Thread for building automation (up to 100m).
3. Long Range: LoRaWAN, Sigfox, NB-IoT for rural agricultural monitoring.
4. Cellular: 4G/5G LTE for mobile vehicle and fleet tracking.

5. Protocols: MQTT (cloud messaging), CoAP (constrained devices), AMQP (queuing).

Q4. Explain the role of key IoT technologies such as sensor technology, RFID technology, and satellite technology.

Textbook Block Diagram:



RFID AND SATELLITE TECHNOLOGY IN IOT

Fig: RFID and Satellite Technology in IoT

1. Sensor Technology:

- Converts physical parameters (temperature, humidity, pressure, light) into electrical signals.
- Forms the Perception Layer — first step of all IoT data collection.
- Examples: DHT11 (temperature), PIR (motion), MQ-2 (gas sensor).

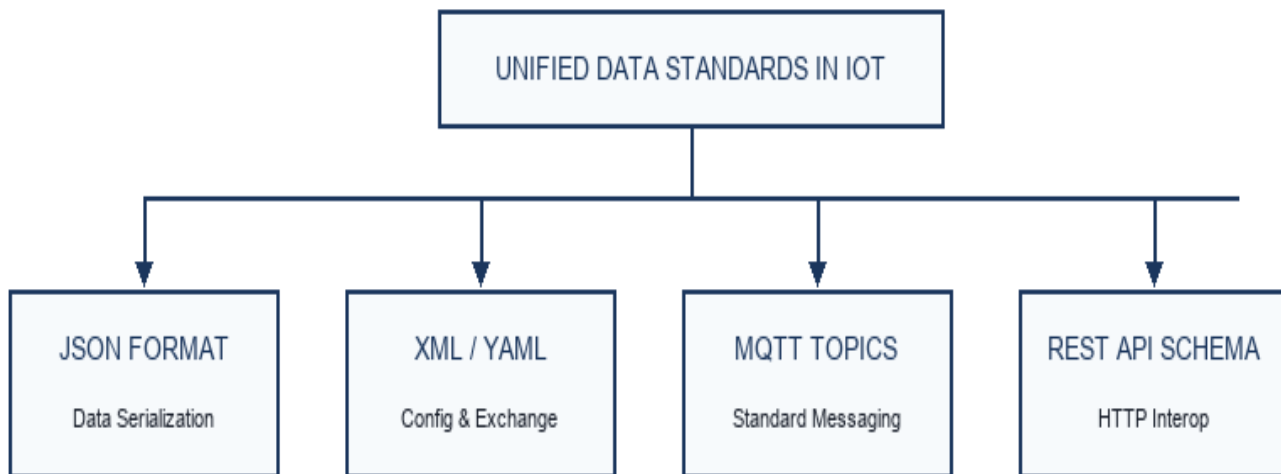
2. RFID Technology:

- Uses radio waves to automatically scan and identify tags on physical objects.
- Three parts: RFID Tag + RFID Reader + Backend Controller.
- Examples: FASTag toll collection, retail inventory, library book management.

3. Satellite Technology:

- Uses LEO satellites to connect remote sensors where cellular networks don't reach.
- Provides 100% global coverage over oceans, mountains, and deserts.
- Examples: Ship cargo tracking, oil pipeline monitoring, wildlife GPS tracking.

Q5. Write about Unified Data Standards. How is it used and what is the importance of it?



UNIFIED DATA STANDARDS ENABLE INTEROPERABILITY ACROSS IOT DEVICES

Fig: Unified Data Standards in IoT Block Diagram

Unified Data Standards are common data formats and protocols (like JSON, XML, MQTT topics) that define HOW IoT devices from different manufacturers share and understand each other's data.

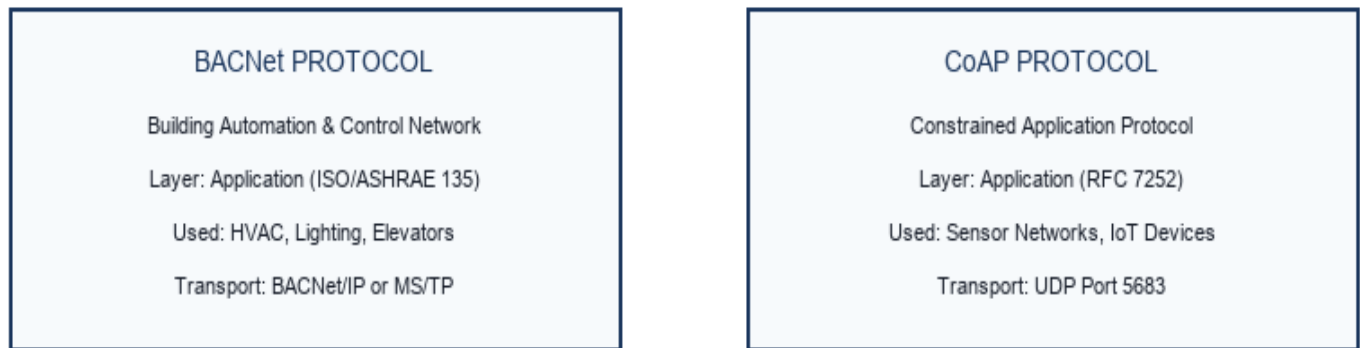
How It Is Used:

1. Data Serialization: JSON format packages sensor readings into lightweight text structures.
2. API Communication: REST APIs use unified JSON/XML schemas for HTTP data exchange.
3. MQTT Topics: Standardized topic naming ensures correct data routing to subscribers.

Importance

1. Interoperability: Allows Philips, Samsung, and Google IoT devices to talk together.
2. Plug & Play: New devices join the network without custom integration code.
3. Reduces Errors: Agreed data schemas prevent misinterpretation of sensor values.
4. Scalability: Adding millions of new devices does not require protocol redesign.

Q6. Write about BACNet Protocol and CoAP.



BACNET AND COAP PROTOCOL COMPARISON BLOCK DIAGRAM

Fig: BACNet and CoAP Protocol Comparison Block Diagram

1. BACNet Protocol (Building Automation and Control Networks):

- Developed by ASHRAE (American Society of Heating, Refrigerating and Air-Conditioning Engineers).
- Standard: ISO 16484-5.
- Purpose: Automates building HVAC systems, lighting, elevators, and fire alarms.
- Transport: BACNet/IP (over Ethernet) or BACNet MS/TP (over RS-485 serial).
- Example: Hotel rooms automatically adjust AC temperature using BACNet thermostats.

2. CoAP Protocol (Constrained Application Protocol):

- Defined in RFC 7252 by IETF (Internet Engineering Task Force).
- Purpose: Lightweight web protocol for 8-bit microcontrollers with limited RAM/ROM.
- Transport: UDP Port 5683 (much faster than TCP-based HTTP).
- Features: RESTful GET/POST/PUT/DELETE methods, Observe push updates, DTLS security.
- Example: Temperature sensors in smart factories send readings via CoAP to edge servers.

Q7. Explain what an IoT System is and write about Raspberry Pi-Board

An IoT system is a complete setup that connects physical devices to the internet so that data can be collected, sent, processed, and used to perform some useful action. A basic IoT system generally works in the following flow:

Block Diagram of a Basic IoT System

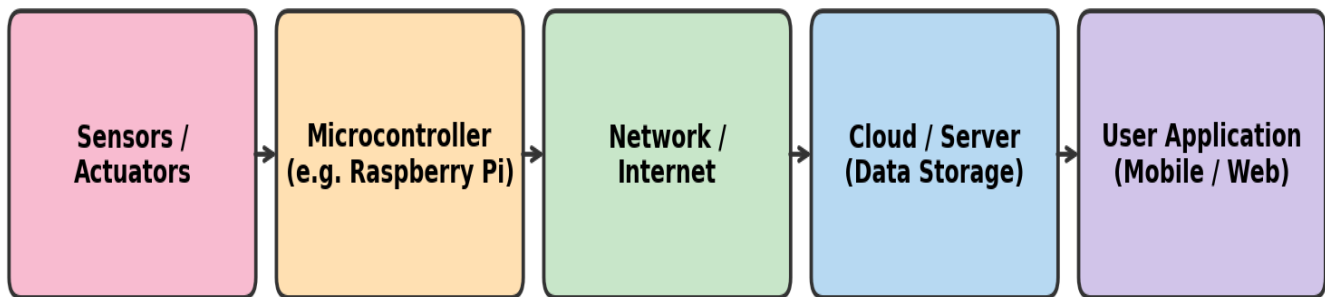
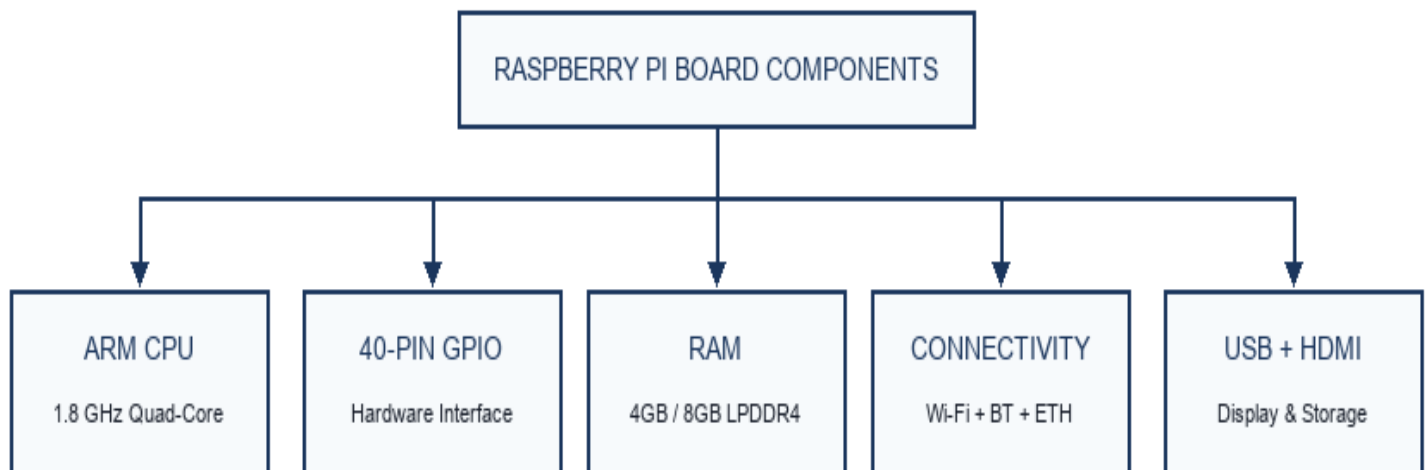


Fig: Block Diagram of a Basic IoT System

Components of an IoT System:

1. Sensors & Actuators (Input/Output hardware devices).
2. IoT Gateway (Local router connecting sensors to internet).
3. Cloud Platform (AWS IoT, Azure for data storage and analytics).
4. User Dashboard (Mobile app or web interface for control).

Raspberry Pi Board:



RASPBERRY PI 4B HARDWARE COMPONENTS BLOCK DIAGRAM

Fig: Raspberry Pi Board Components Block Diagram

Raspberry Pi is a credit-card sized single-board computer (SBC) developed by the Raspberry Pi Foundation for affordable computing education and IoT hardware prototyping.

Key Board Specifications:

1. Processor: ARM Cortex-A72 Quad-Core 1.8 GHz CPU.
2. RAM: 4GB or 8GB LPDDR4 memory.
3. GPIO: 40-pin hardware header for direct sensor interfacing.
4. Connectivity: Built-in Wi-Fi 802.11ac, Bluetooth 5.0, and Gigabit Ethernet.
5. Ports: 2x USB 3.0, 2x USB 2.0, 2x micro-HDMI 4K display outputs.
6. OS: Runs Raspberry Pi OS (Debian Linux-based).

Q8. Write about features and advantages of Raspberry Pi.

Features of Raspberry Pi

- Small in size (credit card size) but works as a full computer.
- Has a powerful ARM based processor, RAM, and GPU on a single board.
- Supports many operating systems, mainly Raspberry Pi OS (Linux based).
- Comes with GPIO pins to connect external hardware like sensors and LEDs.
- Has built-in WiFi, Bluetooth, USB ports, HDMI port, and Ethernet port (in newer models).
- Supports many programming languages such as Python, C, C++, and Java.
- Uses a microSD card as storage instead of a hard disk.

Advantages of Raspberry Pi

- Low cost: Very cheap compared to a normal desktop/laptop computer.
- Low power consumption: Can run for a long time without consuming much electricity.
- Portable: Small size makes it easy to carry and fit into small projects.
- Flexible: Can be used for many different types of projects — IoT, robotics, home automation, media centre, etc.
- Good learning tool: Widely used by students to learn programming, electronics, and IoT practically.

Q9. Explain why Python is preferred in IoT. Write about Characteristics of Python.

Why Python is Preferred in IoT

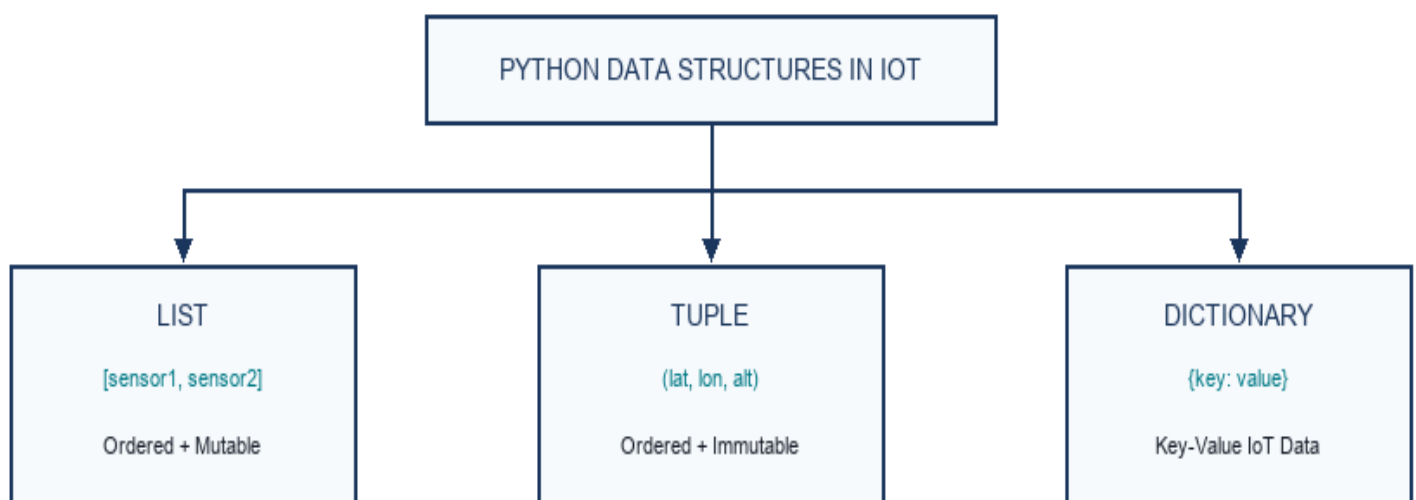
1. Pre-Installed on Raspberry Pi OS: No separate installation needed — ready out of the box.
2. RPi.GPIO Library: Official library to read sensors and control GPIO pins with simple code.
3. Simple & Readable Syntax: Students write hardware scripts in 5-10 lines easily.
4. Massive Ecosystem: Libraries for everything — MQTT (paho-mqtt), sensors (Adafruit), Flask.
5. Cross-Platform: Same Python script runs on Windows, Linux, and Raspberry Pi unchanged.

Key Characteristics of Python:

1. Interpreted Language: Code runs line by line without compilation — easy debugging.
2. Dynamically Typed: No need to declare variable types (`x = 10` just works).
3. Object-Oriented: Supports classes and objects for modular IoT application design.
4. Open Source & Free: Completely free to download, use, and distribute.
5. Large Standard Library: Hundreds of built-in modules (math, datetime, socket, os).
6. Indentation-Based: Uses whitespace indentation instead of curly brackets for code blocks.

Q10. Write about List, Tuple and Dictionary in Python with examples.

Textbook Block Diagram:



LIST, TUPLE AND DICTIONARY IN PYTHON

Fig: Python List, Tuple and Dictionary Data Structures

1. LIST [] — Ordered & Mutable:

- A List stores multiple values in order. Values CAN be changed (mutable).
- Syntax: `variable = [value1, value2, value3]`
- Example in IoT:

```
sensors = ['temperature', 'humidity', 'gas']
sensors.append('motion')    # Adding new sensor
print(sensors[0])          # Output: temperature
```

2. TUPLE () — Ordered & Immutable:

- A Tuple stores multiple values in order. Values CANNOT be changed (immutable).
- Syntax: `variable = (value1, value2, value3)`
- Example in IoT:

```
location = (27.7172, 85.3240) # GPS: Latitude, Longitude (fixed values)
print(location[0])           # Output: 27.7172
```

3. DICTIONARY { } — Key:Value Pairs:

- A Dictionary stores data as key:value pairs. Values accessed by name (key).
- Syntax: `variable = {'key1': value1, 'key2': value2}`
- Example in IoT:

```
sensor_data = {'temperature': 32, 'humidity': 75, 'gas': 'normal'}
print(sensor_data['temperature']) # Output: 32
sensor_data['gas'] = 'high'       # Updating value
```

Feature	List	Tuple	Dictionary
Brackets	[]	()	{ }
Ordered	Yes	Yes	No (Key-Based)
Mutable	Yes — can change	No — fixed	Yes — can change
Access By	Index [0]	Index [0]	Key ['name']
IoT Use	Sensor name list	Fixed GPS coords	Sensor readings data

Q1. What are the characteristics of IoT? Explain with examples.

The Internet of Things (IoT) is a global network of physical objects embedded with sensors, software, and internet connectivity that automatically collect, exchange, and act on data without human involvement.

Key Characteristics of IoT (5 Points):

1. Interconnectivity:

Any physical device can connect to the internet and communicate with other devices.

Example: Smart TV, smart fridge, and smart AC all connected to one home Wi-Fi network.

2. Unique Identity:

Every IoT device has a unique IP address or MAC address for identification.

Example: Each smart meter has a unique IPv6 address to report electricity data.

3. Dynamic & Self-Adapting:

Devices automatically change behavior based on real-time environment conditions.

Example: Smart thermostat increases AC cooling when room temperature rises above 30°C.

4. Sensing & Actuation:

Captures real-world physical inputs (sensing) and triggers physical actions (actuation).

Example: Motion sensor detects intruder → automatically activates alarm siren.

5. Massive Scalability:

IoT supports billions of connected sensor nodes across the global internet.

Example: Smart city with thousands of traffic sensors and pollution monitors.

Q2. What are the benefits of IoT devices?

Main Benefits of IoT Devices (

1. Remote Monitoring & Control:

Users can monitor and control IoT devices from anywhere using smartphone apps.

2. Automation of Tasks:

Machines work automatically without manual human operation, saving time.

3. Cost & Energy Savings:

Smart meters and smart lighting reduce electricity wastage significantly..

4. Improved Safety & Healthcare:

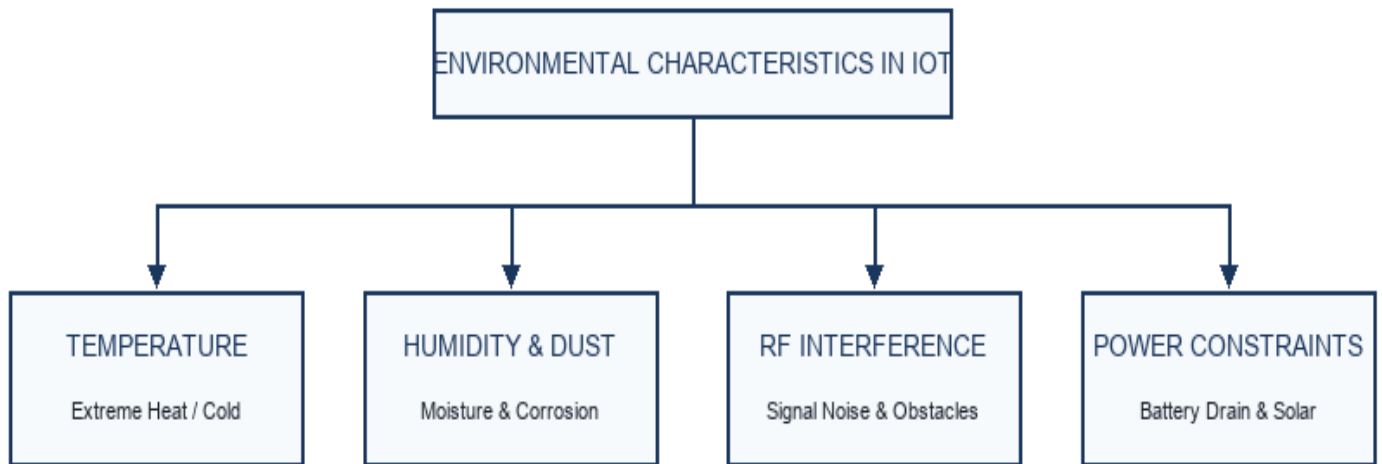
Wearable sensors detect health emergencies and send instant alerts..

5. Real-Time Data & Better Decisions:

Businesses use real-time IoT data analytics to optimize operations.

Q3. What do you mean by environment characteristics of IoT? Explain.

Textbook Block Diagram:



MAIN ENVIRONMENTAL FACTORS IMPACTING IOT PERFORMANCE

Fig: Environmental Characteristics Impacting IoT Performance

Environmental characteristics are the physical and real-world external conditions surrounding IoT hardware — like temperature, humidity, dust, and signal interference — that directly affect how well IoT devices and wireless networks perform.

Key Environmental Characteristics

1. Extreme Temperature Variations:

Cold freezing winters or scorching summer heat can degrade battery efficiency and reduce sensor measurement accuracy.

2. Moisture, Humidity & Dust:

High humidity and dust cause short-circuits and corrosion in outdoor sensor circuit boards.

3. RF Signal Interference & Obstacles:

Physical walls, metal beams, and electrical equipment block wireless Wi-Fi and Zigbee radio signals.

4. Power & Energy Constraints:

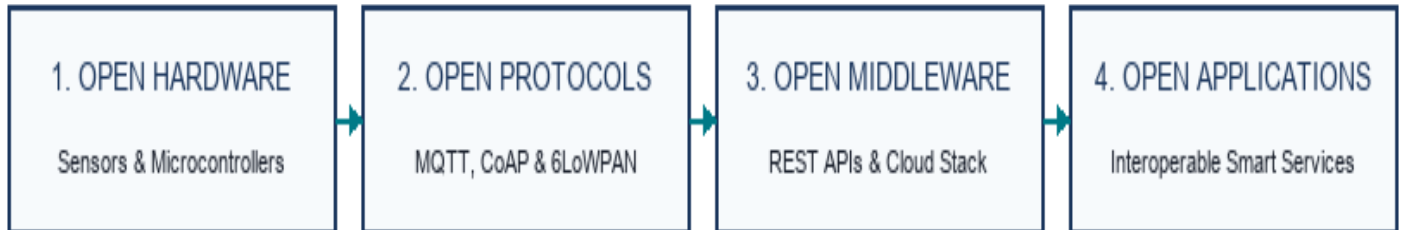
Remote outdoor sensors relying on small batteries or solar panels face constant power depletion.

5. Physical Vibration & Shock:

Heavy industrial machinery vibrations physically damage delicate sensor pins and solder joints.

Q4. Define open architecture. Explain with neat diagram.

Open Architecture in IoT is a modular, vendor-neutral system design based on publicly available open standards and APIs, allowing devices and software from any manufacturer to freely integrate and communicate without compatibility issues.



OPEN IOT ARCHITECTURE BLOCK DIAGRAM

Fig: Open IoT Architecture Block Diagram

Features of Open Architecture

1. Open Standards: Uses publicly available protocols like MQTT, REST, JSON, and IPv6.
2. No Vendor Lock-In: Organizations are not forced to buy from a single company.
3. Interoperability: Mix hardware and software from different manufacturers freely.
4. Easy Scalability: Add new devices and upgrade components without rebuilding the system.
5. Community Driven: Open-source communities continuously improve and maintain the platform.

Benefits:

- Reduces costs by allowing competitive hardware purchasing.
- Enables future-proof system upgrades as technology evolves.

Q5. What do you mean by RFID technology? Explain it.

RFID (Radio Frequency Identification) is a wireless identification technology that

uses radio frequency signals to automatically detect, identify, and track physical objects attached with electronic tags without requiring line-of-sight contact.

Three Core Components of RFID:

1. RFID Tag (Transponder): A tiny chip with antenna attached to the object. Stores the unique ID data.
2. RFID Reader (Interrogator): Sends radio wave signals to power and read the tag's stored ID.
3. Backend Controller System: Receives tag data from reader and processes it in the database.

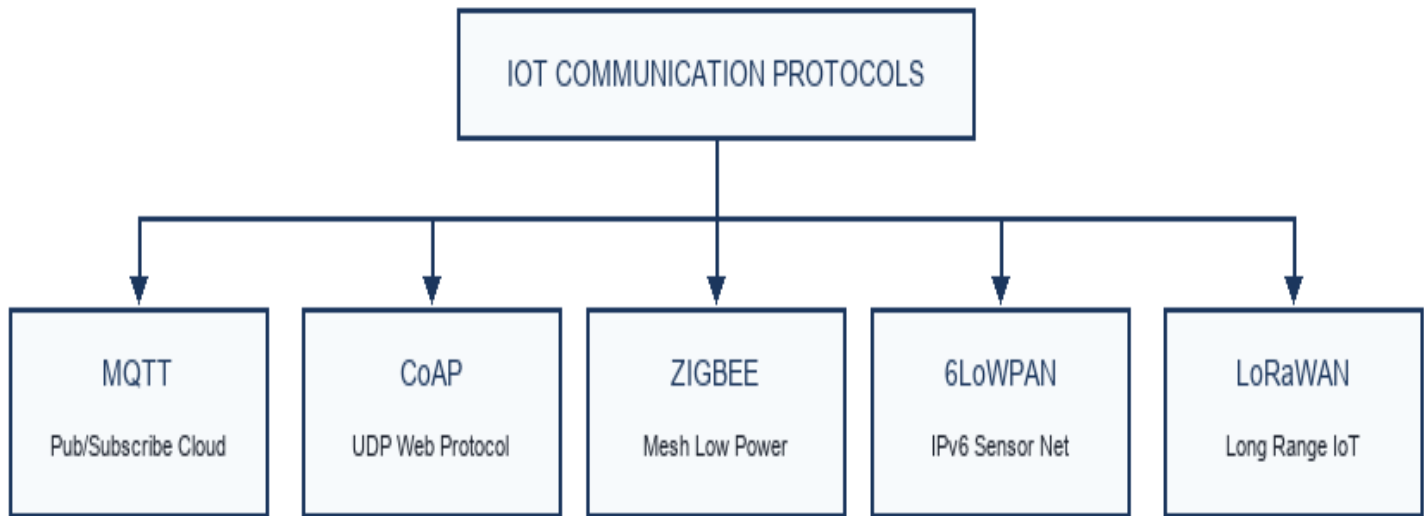
Types of RFID Tags:

- Passive Tag: No internal battery. Powered wirelessly by reader signal. Range: up to 10 meters.
- Active Tag: Has internal battery. Continuously broadcasts signal. Range: up to 100 meters.

Real-World Applications:

1. FASTag Toll Collection: Cars pass toll booths without stopping — RFID tag scanned automatically.
2. Retail Inventory: Warehouse workers scan thousands of items in seconds using RFID readers.
3. Hospital Patient Wristbands: RFID wristband stores patient ID for quick staff identification.
4. Library Book Tracking: Students borrow books — RFID system updates library records instantly.

Q6. What do you mean by protocols in IoT? Explain any.



MAJOR IOT COMMUNICATION PROTOCOLS OVERVIEW

Fig: IoT Communication Protocols Overview

IoT Protocols are standardized sets of communication rules that define HOW IoT devices send, receive, and understand data across networks — ensuring reliable, secure, and compatible data exchange between sensors, gateways, and cloud servers.

Why Protocols Are Important:

- Without protocols, devices from different brands cannot understand each other's data.
- Protocols ensure data arrives correctly, securely, and in the right order.

MQTT Protocol (Explained in Detail):

Full Name: Message Queuing Telemetry Transport.

Architecture — How MQTT Works:

Publisher (Sensor) → MQTT Broker Server → Subscriber (Dashboard/App)

Key Features of MQTT:

1. Publish-Subscribe Model: Sensors publish data to topics; dashboards subscribe to those topics.
2. Transport: Runs over TCP/IP on Port 1883 (or 8883 with TLS encryption).

3. Extremely Lightweight: Only 2-byte minimum header — ideal for slow 2G mobile networks.
4. QoS Levels: Guarantees message delivery at 3 levels (0=fire-forget, 1=at-least-once, 2=exactly-once).
5. Used By: AWS IoT Core, Google Cloud IoT, Home Assistant smart home hubs.

Example:

A Raspberry Pi temperature sensor publishes `{temp: 32}` to topic `home/bedroom/temp`.

Mobile app subscribed to same topic receives alert when temperature exceeds 35°C.



Q7. What do you mean by Pi? Is there any difference between general purpose computer and Pi (Raspberry Pi)?

Definition of Raspberry Pi (2 Lines):

Raspberry Pi is a credit-card sized, affordable single-board computer (SBC) developed by the Raspberry Pi Foundation in the UK, designed for computer education, electronics prototyping, and IoT hardware control projects.

Textbook Comparison Table:

Feature	General Purpose Computer (PC/Laptop)	Raspberry Pi (Single Board Computer)
Size & Form	Large desktop tower or laptop chassis	Tiny credit-card sized single PCB board
GPIO Hardware Pins	No exposed GPIO hardware pins	40-Pin GPIO header for direct sensors
Processor Type	High-power Intel/AMD x86/x64 CPU	Low-power ARM Cortex-A CPU
Operating System	Windows 11 or macOS	Raspberry Pi OS (Debian Linux)
Power Consumption	65W to 500W from AC mains power	Only 5V USB-C, 3W to 15W maximum
Price	\$300 to \$1500+ range	\$35 to \$75 only
RAM	8GB to 64GB DDR5	1GB to 8GB LPDDR4
IoT Suitable?	No — no GPIO, high power, expensive	Yes — GPIO, low power, cheap, sensors

Q8. Write short notes on: (Any Two) — a. 6LoWPAN b. RPi GPIO c. Sensor Technology

a. 6LoWPAN — Short Note

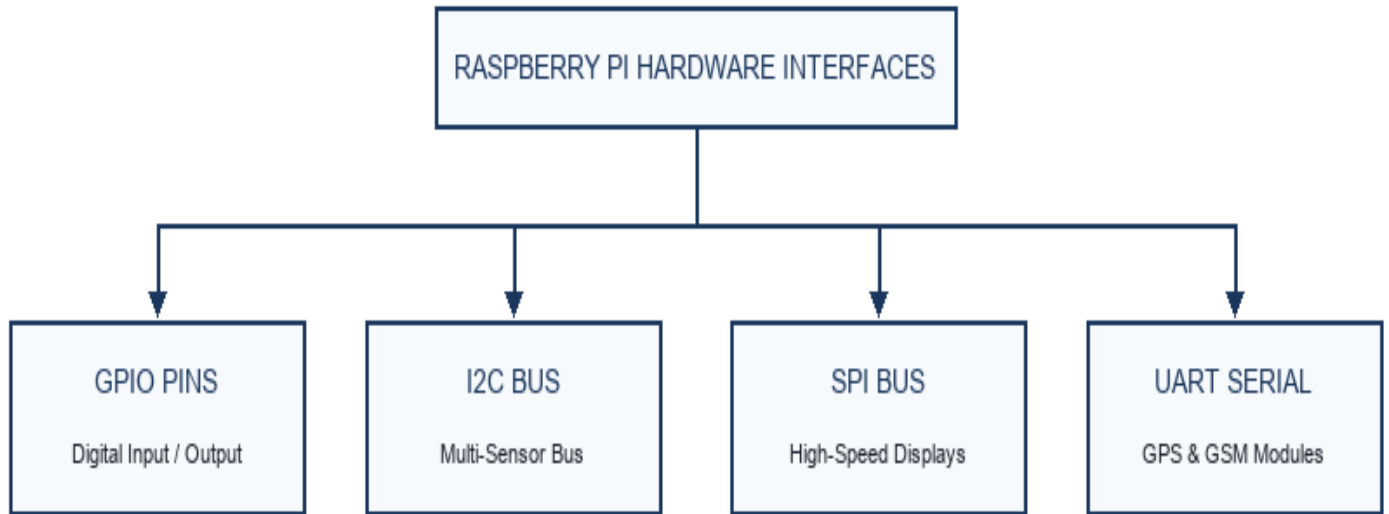
Definition (2 Lines):

6LoWPAN (IPv6 over Low-Power Wireless Personal Area Networks) is an adaptation layer protocol that enables IPv6 internet addressing for battery-powered sensors connected via IEEE 802.15.4 wireless radio networks.

Key Points

1. Enables small sensor nodes to have real IPv6 internet addresses directly.
2. Performs IPv6 header compression — reduces 40-byte headers down to just 2-3 bytes.
3. Fragments large IPv6 packets to fit inside small 127-byte IEEE 802.15.4 frames.
4. Supports mesh routing using RPL (IPv6 Routing Protocol for Low-Power Networks).
5. Bridges low-power sensor sub-networks directly to the global internet without NAT.
6. Used in smart electricity metering, building automation, and body area networks.
7. Key networking standard at the foundation of the modern Internet of Things.

b. RPi GPIO — Short Note



RASPBERRY PI HARDWARE INTERFACES FOR IOT APPLICATIONS

Fig: Raspberry Pi GPIO and Hardware Interfaces

Definition (2 Lines):

Raspberry Pi GPIO (General Purpose Input/Output) refers to the 40 programmable physical pins on the Raspberry Pi board used to directly interface with external hardware sensors, LEDs, relays, and communication modules.

Key Points (7 Points):

1. Contains 26 digital GPIO pins operating at 3.3V safe logic voltage level.
2. Provides dedicated 5V, 3.3V power supply pins and multiple Ground (GND) pins.
3. Controlled easily with Python using official `RPi.GPIO` or modern `gpiozero` libraries.
4. Supports I2C (data+clock) bus for multi-sensor communication.
5. Supports SPI (MOSI/MISO/CLK/CS) bus for fast LCD display communication.
6. Supports UART (TX/RX) serial pins for GPS and GSM modem connection.
7. Most important Raspberry Pi feature for building real-world IoT hardware projects.

c. Sensor Technology — Short Note

Definition (2 Lines):

Sensor Technology in IoT refers to electronic devices that detect and measure physical real-world parameters (temperature, humidity, motion, pressure, gas) and convert them into electrical signals for digital processing.

Key Points (7 Points):

1. Detects parameters: temperature, humidity, pressure, gas levels, motion, and light intensity.
2. Active Sensors need external power supply; Passive Sensors self-generate their signals.
3. Outputs either analog voltage (0-5V) or digital protocol data (I2C, SPI, or UART).
4. MEMS (Micro-Electro-Mechanical Systems) miniaturizes sensors onto tiny silicon microchips.
5. Key design goals: High accuracy, low power consumption, small size, and low cost.
6. Used in smart agriculture, wearable healthcare, industrial automation, and smart cities.
7. Forms the Perception Layer — the foundational physical data

Anyad Ali
PUBLISHER

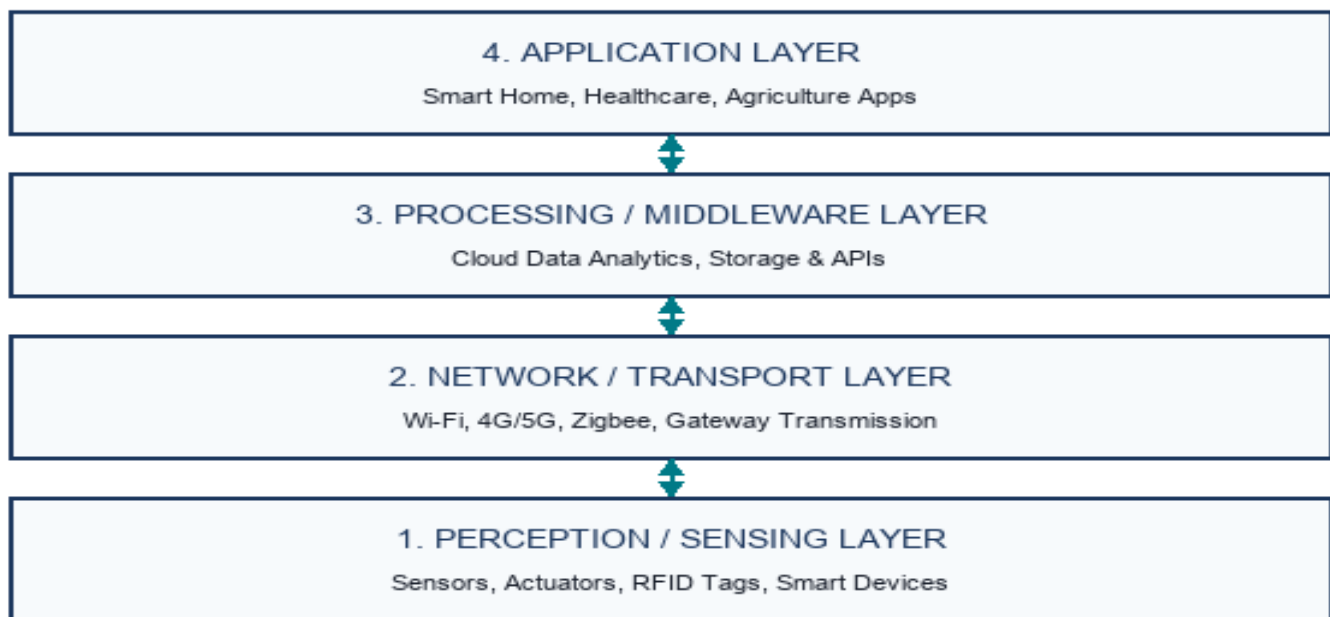
IoT VVI Model Questions — UNIT 1

Introduction to Internet of Things (IoT)

CTEVT Diploma IT/DCOM | Simple Student Language + Block Diagrams

Q1. Define IoT. Explain the architecture of IoT with a neat diagram.

The Internet of Things (IoT) is a global network of physical objects embedded with sensors, software, and internet connectivity that automatically collect, exchange, and act on real-world data without human intervention.



4-LAYER IOT ARCHITECTURE BLOCK DIAGRAM

Fig: 4-Layer IoT Architecture

Four Layers Explained:

1. Perception / Sensing Layer: Physical sensors (temperature, gas, motion) and actuators collect real-world data.
2. Network / Transport Layer: Transmits data wirelessly via Wi-Fi, 4G/5G, Zigbee, Bluetooth, LoRaWAN.
3. Processing / Middleware Layer: Cloud servers store, clean, and analyze incoming sensor data.
4. Application Layer: Delivers smart dashboards, alerts, and controls to end users (Smart Home, Healthcare).

Q2. Explain the history and evolution of IoT.

History & Evolution of IoT (Timeline):

1. 1982 — First Internet-connected machine: A Coca-Cola vending machine at Carnegie Mellon University reported its stock and drink temperature online.
2. 1990 — First IoT Device: John Romkey connected a toaster to the internet at INTEROP conference.
3. 1999 — Term 'IoT' Coined: Kevin Ashton (MIT Auto-ID Center) first used 'Internet of Things' while working on RFID supply chain systems.
4. 2000 — LG Smart Fridge: LG launched the first Internet-enabled refrigerator that could track food stock.
5. 2008-2009 — IoT Birth: Number of internet-connected devices exceeded the world's human population.
6. 2010s — Smart Phones + Cloud: Rise of smartphones, AWS cloud, and affordable sensors accelerated IoT adoption.
7. 2020s — Billion Devices: Over 15 billion IoT devices globally; AI + 5G driving next-gen smart cities and healthcare.



Q3. Discuss any two IoT frameworks with their features.

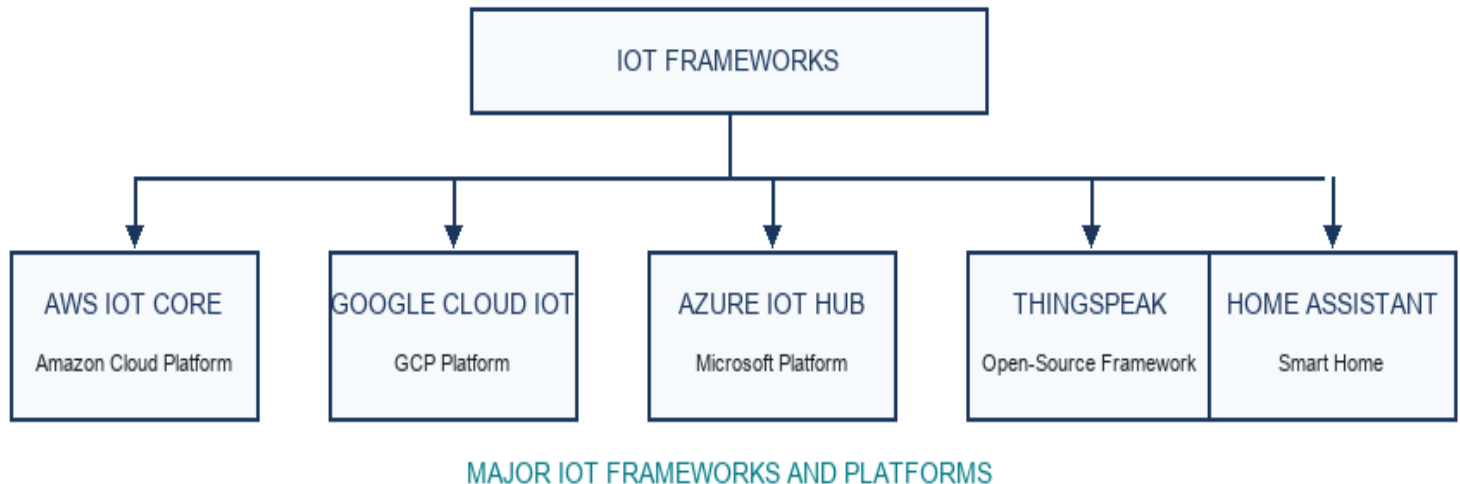


Fig: Major IoT Frameworks

1. AWS IoT Core (Amazon Web Services):

- Full managed cloud service for connecting millions of IoT devices securely.
- Uses MQTT and HTTP protocols for device messaging.
- Features: Device Registry, Device Shadow, Rules Engine, OTA updates.
- Integrates with Lambda (serverless), DynamoDB (database), and S3 (storage).
- Best for: Large-scale enterprise industrial IoT deployments.

2. Microsoft Azure IoT Hub:

- Cloud gateway that enables secure bidirectional communication with IoT devices.
- Supports MQTT, AMQP, and HTTPS protocols.
- Features: Device Twins (digital shadows), Direct Methods, OTA firmware updates.
- Integrates with Power BI for rich dashboard analytics and visualization.
- Best for: Enterprise Windows-based IoT and industrial monitoring systems.

Q4. Differentiate between IoT and M2M.

Feature	IoT (Internet of Things)	M2M (Machine-to-Machine)
Full Form	Internet of Things	Machine to Machine

Communication	Devices communicate via Internet/Cloud	Direct device-to-device link
Intelligence	AI/ML cloud analytics applied	Limited local intelligence
Scalability	Billions of devices globally	Limited connections
Protocols	MQTT, CoAP, HTTP REST	Proprietary or cellular
Human Interaction	Mobile app dashboards	Minimal user interaction
Example	Smart home ecosystem	Industrial equipment telemetry

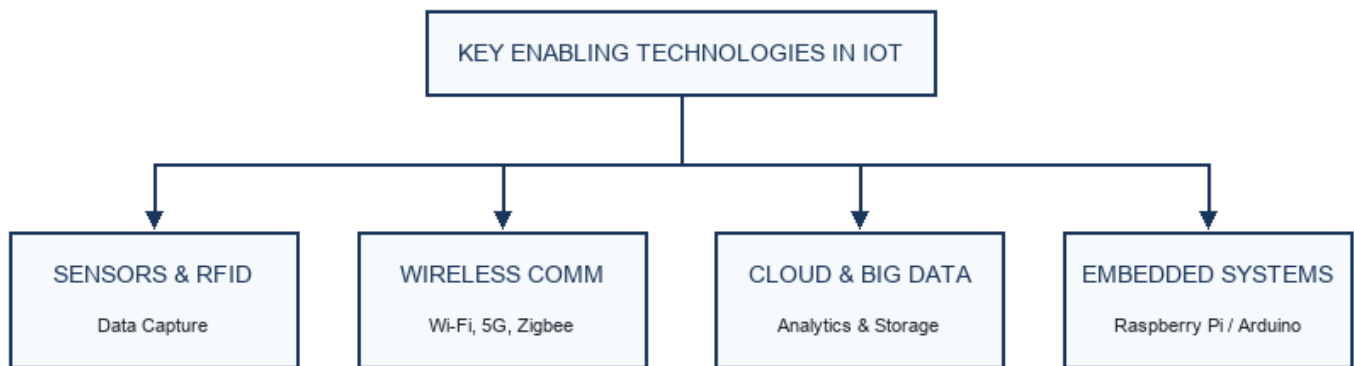
Q5. List and explain the layers of IoT architecture (3-layer vs 5-layer model).

Layer #	3-Layer Model	5-Layer Model
Layer 1	Perception Layer	Perception Layer (Sensors)
Layer 2	Network Layer	Transport Layer (Wireless Comm)
Layer 3	Application Layer	Processing Layer (Cloud Analytics)
Layer 4	—	Application Layer (User Apps)
Layer 5	—	Business Layer (Business Logic & Decisions)

IoT VVI Model Questions — UNIT 2

Fundamental Mechanisms and Key Technologies
CTEVT Diploma IT/DCOM | Simple Student Language + Block Diagrams

Q1. Explain the key technologies used in IoT (RFID, sensor technology, communication protocols, etc)



CORE TECHNOLOGIES ENABLING FUNCTIONING OF IOT

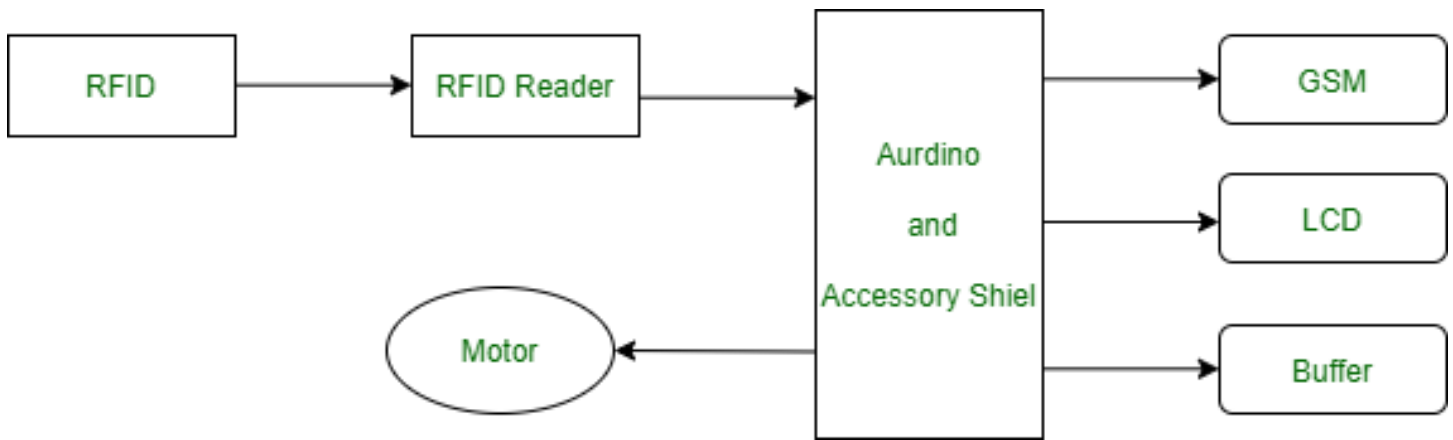
Fig: Key Enabling Technologies in IoT

Key Technologies in IoT.

1. **RFID Technology:** Uses radio waves to automatically identify and track tagged physical objects without line-of-sight contact.
2. **Sensor & Actuator Technology:** Sensors convert physical parameters (temperature, motion, gas) to electrical signals. Actuators perform physical actions (relays, motors).
3. **Wireless Communication Protocols:** Enable data transmission between devices — Wi-Fi, Zigbee, Bluetooth BLE, 4G/5G, LoRaWAN.
4. **Cloud & Edge Computing:** Cloud provides scalable data storage and ML analytics; Edge computing processes data locally for low latency.

Q2. What is RFID? Explain its working principle and applications in IoT.

RFID (Radio Frequency Identification) is a wireless identification technology that uses radio frequency signals to automatically detect and identify electronic tags attached to physical objects



Working Principle: diagram

Working Principle:

Step 1: RFID Reader emits radio frequency electromagnetic waves.

Step 2: Passive Tag receives the signal, powers up its chip, and transmits its unique stored ID back.

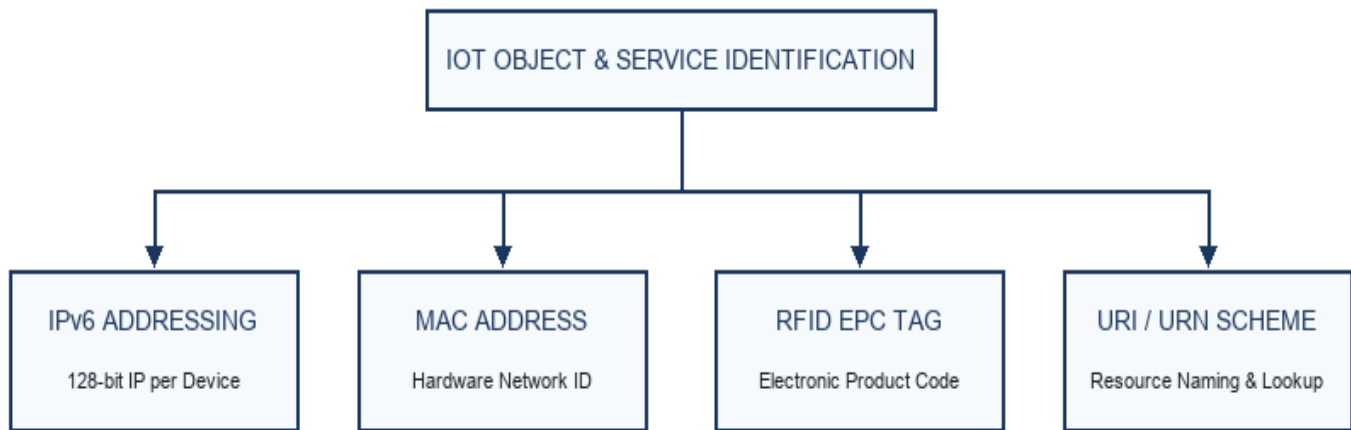
Step 3: RFID Reader receives the tag ID and sends it to the backend database controller.

Step 4: Controller matches the received ID with the database to confirm identity.

Applications in IoT:

1. FASTag Toll Collection: Vehicles pass toll booths without stopping — RFID scanned automatically.
2. Smart Retail Inventory: Scan thousands of products simultaneously in warehouse.
3. Hospital Patient ID: RFID wristbands identify patients and their medical records instantly.
4. Library Management: Track book borrow/return automatically.
5. Animal Tracking: RFID chips implanted in livestock for farm inventory management.

Q3. Discuss the structural aspects and identification of IoT objects/services.



PROPER IDENTIFICATION ENHANCES EFFICIENT ROUTING & SECURITY

Fig: IoT Object & Service Identification

Structural Aspects of IoT Objects:

1. Hardware Layer: Physical sensor/actuator devices with embedded microcontrollers.
2. Communication Layer: Wireless transceivers enabling connectivity (Wi-Fi, BLE, Zigbee).
3. Software Layer: Firmware, OS, and application software managing device behavior.

Identification Methods:

1. IPv6 Addressing: 128-bit unique IP address assigned to each IoT device globally.
2. MAC Address: Hardware-level 48-bit unique identifier burned into network interface chip.
3. RFID EPC (Electronic Product Code): Unique tag ID for physical object identification.
4. URI / URN Scheme: Web-based naming standard to uniquely identify IoT services.

Importance of Proper Identification:

- Enables precise data routing to correct cloud server.
- Blocks unauthorized rogue devices from network access.
- Allows service discovery and dynamic device registration.

Q4. Explain scalability and interoperability challenges in IoT.

1. Scalability Challenges:

- Billions of concurrent device connections stress cloud server infrastructure.
- Network bandwidth bottlenecks when thousands of sensors transmit simultaneously.
- Database storage and query performance degrades at massive IoT data scale.
- Solution: Edge computing reduces cloud load by processing data locally.

2. Interoperability Challenges:

- Devices from different manufacturers use different proprietary protocols.
- Multiple fragmented standards: Zigbee, Z-Wave, Thread, Wi-Fi all incompatible.
- Cloud platforms (AWS, Azure, Google) use different API formats.
- Solution: Open standards (MQTT, JSON, IPv6) and unified middleware gateways.

Q5. Describe the role of sensor technology in IoT with examples.

Role of Sensor Technology in IoT:

Sensors form the Perception Layer — the first physical data entry point in all IoT systems. They detect real-world environmental changes and convert them to digital signals.

Key Sensor Types & Examples:

1. Temperature Sensor (DHT11/DS18B20): Reads ambient temperature — used in smart AC, cold chain logistics.
2. Motion Sensor (PIR HC-SR501): Detects human movement — used in smart security alarms.
3. Gas Sensor (MQ-2, MQ-135): Detects LPG/smoke/CO2 levels — used in smart kitchen safety systems.
4. Humidity Sensor (DHT22): Measures air moisture — used in smart greenhouse cultivation.
5. Pressure Sensor (BMP280): Measures atmospheric pressure — used in weather monitoring drones.
6. Ultrasonic Sensor (HC-SR04): Measures distance — used in smart parking and robot obstacle detection.

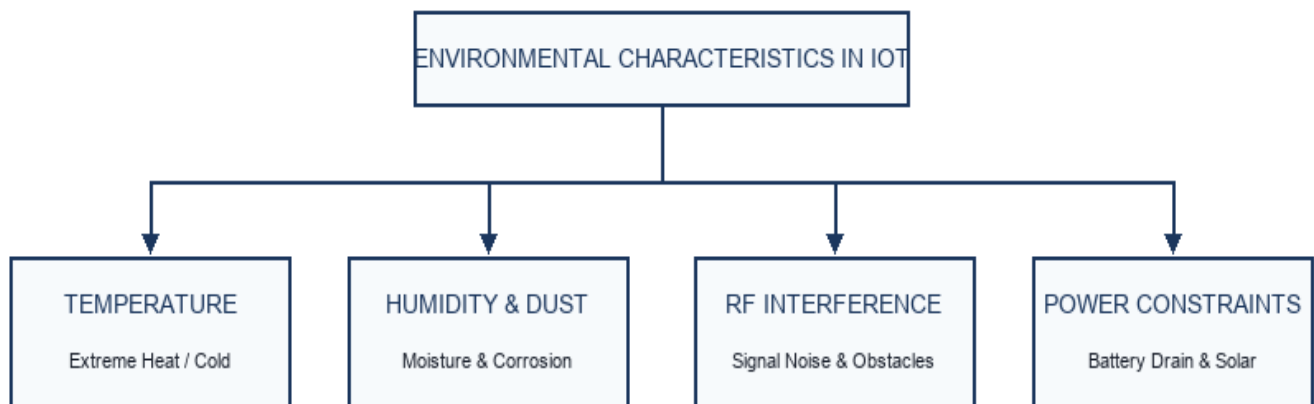
Q6. Explain satellite technology's role in IoT communication.

Satellite IoT uses Low Earth Orbit (LEO) satellite constellations to provide wireless connectivity to IoT sensors in remote locations where cellular towers and Wi-Fi don't exist.

Role in IoT Communication.

1. Global Coverage: 100% coverage over oceans, mountains, deserts, and polar regions.
2. Remote Monitoring: Oil pipeline pressure sensors in Sahara desert transmit via satellite.
3. Maritime Tracking: Ships in ocean report GPS cargo location and engine data via satellite.
4. Environmental Monitoring: Weather buoys in Pacific Ocean send ocean temperature data.
5. Disaster Relief: Deploys instant connectivity for emergency response in earthquake zones.

Q7. What are the environment and traffic characteristics of IoT.



MAIN ENVIRONMENTAL FACTORS IMPACTING IOT PERFORMANCE

Fig: Environmental Characteristics in IoT

Environmental Characteristics:

1. Extreme Temperature: Heat/cold degrades battery and sensor accuracy.
2. Moisture & Humidity: Causes corrosion and short-circuits in outdoor sensors.
3. RF Signal Interference: Walls, metal, and electricity block wireless signals.

4. Power Constraints: Remote sensors rely on batteries/solar with limited energy.

Traffic Characteristics:

1. Bursty Traffic: Many sensors transmit simultaneously causing network spikes.
2. Low Latency Required: Medical/industrial IoT needs real-time <100ms response.
3. Small Packet Sizes: Sensor readings are tiny (10-100 bytes) but very frequent.

Q8. Differentiate between active and passive RFID. [Short —

Feature	Active RFID Tag	Passive RFID Tag
Power Source	Internal battery (self-powered)	No battery — powered by reader signal
Range	Up to 100 meters	Up to 10 meters only
Cost	More expensive	Very cheap
Size	Larger (battery inside)	Very small and thin
Lifespan	Battery life (3-5 years)	Permanent — no battery to replace
Application	Vehicle tracking, asset management	FASTag, library books, access cards

Q9. Explain Open Architecture in IoT and Security & Privacy issues in IoT.



OPEN IOT ARCHITECTURE BLOCK DIAGRAM

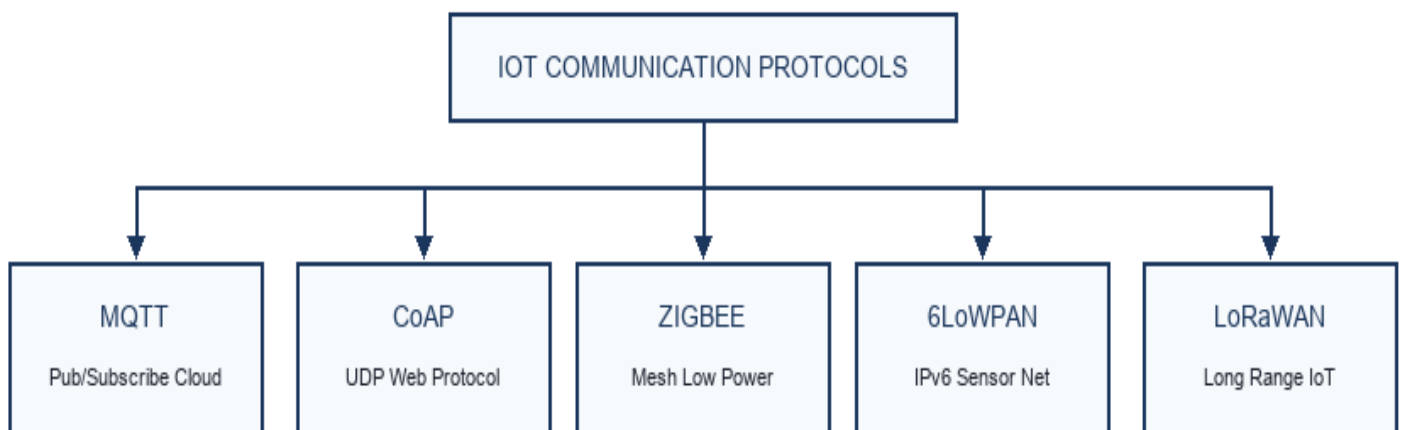
Fig: Open IoT Architecture

Open Architecture: A vendor-neutral IoT design using public standards (MQTT, REST, JSON, IPv6) allowing any manufacturer's hardware/software to freely integrate.

Security & Privacy Issues in IoT:

1. Weak Passwords: Default credentials allow easy hacking.
2. Unencrypted Data: Sensor data transmitted without TLS is intercepted easily.
3. No Firmware Updates: Unpatched bugs leave permanent security holes.
4. Data Privacy: Personal health/location data collected without user consent.
5. Physical Tampering: Exposed GPIO pins allow direct memory key extraction.

Q10. Explain communication capabilities required for IoT devices



MAJOR IOT COMMUNICATION PROTOCOLS OVERVIEW

Fig: IoT Communication Protocols

Communication Capabilities Required:

1. Short Range (In-Room): Bluetooth BLE (10m), Zigbee (100m mesh), Z-Wave (30m).
2. Medium Range (Building): Wi-Fi 802.11 (50-100m), Thread, 6LoWPAN.
3. Long Range (Outdoor Field): LoRaWAN (15km), Sigfox (50km), NB-IoT.
4. Cellular (Mobile): 4G LTE and 5G for mobile fleet and vehicle tracking.
5. Protocols: MQTT (cloud messaging), CoAP (constrained devices), AMQP (queuing).

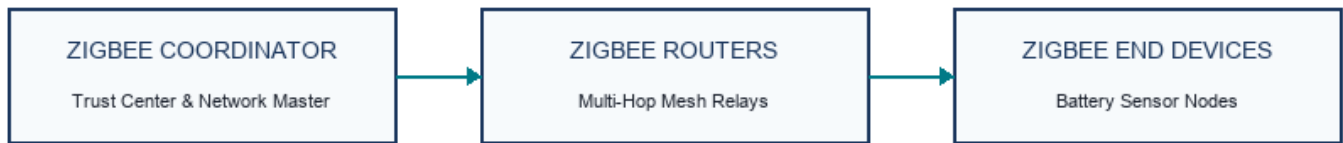


IoT VVI Model Questions — UNIT 3

IoT Protocols

CTEVT Diploma IT/DCOM | Simple Student Language + Block Diagrams

Q1. Explain the Zigbee architecture with its protocol stack



ZIGBEE MESH NETWORK ARCHITECTURE BLOCK DIAGRAM

Fig: Zigbee Mesh Network Architecture

Zigbee is a low-power, low-cost wireless mesh networking protocol based on IEEE 802.15.4 standard, designed for short-range device communication in smart homes and industrial automation.

Zigbee Protocol Stack (Layers):

1. Physical Layer (PHY): Radio transmission at 2.4 GHz (250 kbps) based on IEEE 802.15.4.
2. MAC Layer: Manages channel access using CSMA-CA collision avoidance.
3. Network Layer (NWK): Handles mesh routing and device address management.
4. Application Support Layer (APS): Manages device profiles and endpoint bindings.
5. Application Layer (APL): User application logic and ZigBee Device Objects (ZDO).

Three Device Types:

- Zigbee Coordinator (ZC): Network master — forms network and holds security keys.
- Zigbee Router (ZR): Relay node — extends range by forwarding packets in mesh.
- Zigbee End Device (ZED): Battery leaf node — only talks to its parent router.

Q2. What is CoAP? Explain its message model and how it differs from HTTP.

CoAP (Constrained Application Protocol) is a lightweight web transfer protocol designed for resource-constrained 8-bit microcontrollers and low-power lossy networks, operating over UDP.

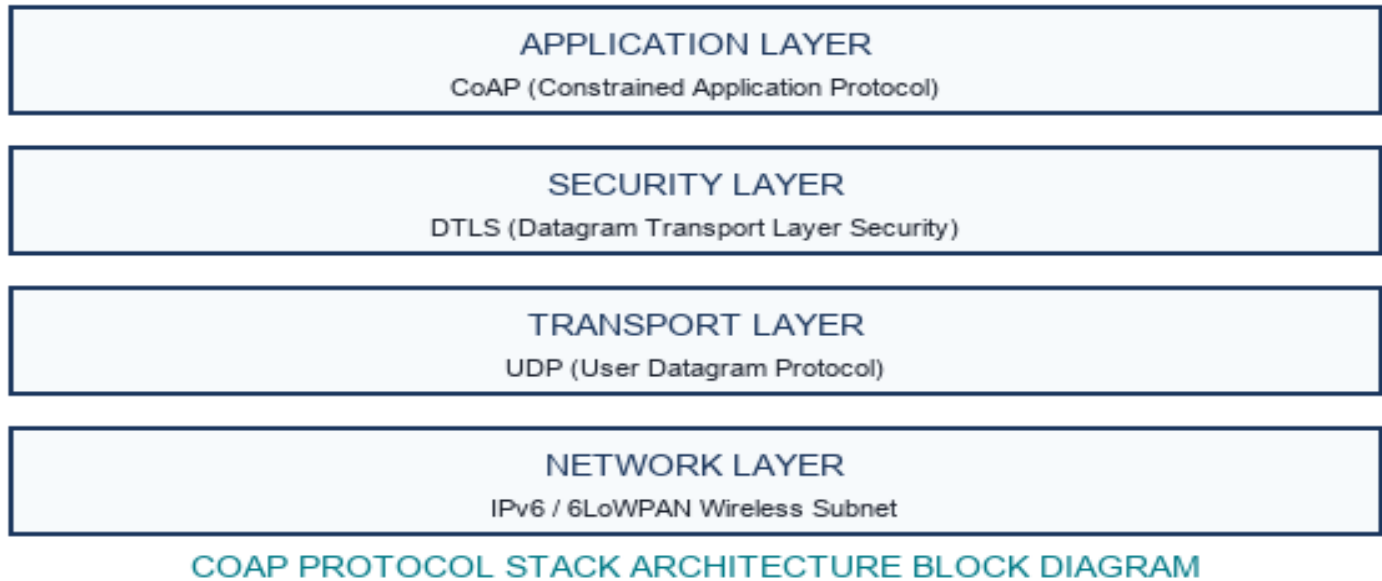


Fig: CoAP Protocol Stack Architecture

CoAP Message Model:

1. CON (Confirmable): Message requiring ACK acknowledgment — reliable delivery.
2. NON (Non-Confirmable): Fire-and-forget message — no ACK needed.
3. ACK (Acknowledgment): Reply confirming receipt of CON message.
4. RST (Reset): Indicates message cannot be processed.

CoAP Features: GET, POST, PUT, DELETE (REST methods), Observe mode (push updates), DTLS security.

Feature	CoAP	HTTP
Transport	UDP (unreliable but fast)	TCP (reliable but heavy)
Header Size	4 bytes (ultra compact)	200+ bytes (large overhead)
Port	5683 (CoAP) / 5684 (DTLS)	80 (HTTP) / 443 (HTTPS)
Security	DTLS encryption	TLS/SSL encryption
Power Usage	Ultra-low power	High power consumption
Best For	8-bit microcontrollers, sensors	Web browsers, servers

Q3. Explain IEEE 802.15.4 standard and its significance in IoT.

IEEE 802.15.4 is an international wireless standard that defines the Physical (PHY) and MAC layers for Low-Rate Wireless Personal Area Networks (LR-WPAN) used by battery-powered IoT sensors.

Key Technical Specifications:

- Frequency: 2.4 GHz global ISM band (also 868 MHz EU, 915 MHz US).
- Data Rate: 250 kbps (sufficient for periodic sensor readings).
- Range: 10m to 100m depending on antenna and environment.
- Channel Access: CSMA-CA (Carrier Sense Multiple Access with Collision Avoidance).
- Topology: Star or Peer-to-Peer mesh networks.
- Power: Ultra-low duty cycle — sensors sleep 99% of time for years on coin batteries.

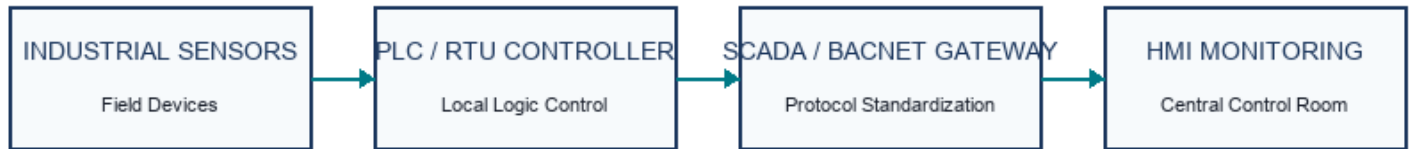
Significance in IoT:

1. Foundation standard for Zigbee, 6LoWPAN, WirelessHART, and Thread protocols.
2. Enables low-cost mass deployment of wireless sensor networks.
3. Standardizes PHY/MAC so upper protocol layers remain interchangeable.

Q4. Compare M2M and WSN protocols.

Feature	M2M Protocol	WSN Protocol
Full Form	Machine to Machine	Wireless Sensor Network
Communication	Point-to-point direct link	Multi-hop mesh network
Network Scale	Few device connections	Thousands of sensor nodes
Power Source	Grid power or heavy battery	Ultra-low coin batteries
Primary Goal	Remote equipment control	Environmental data gathering
Protocols Used	MQTT, CoAP, TR-069, OMA-LWM2M	Zigbee (802.15.4), 6LoWPAN, Z-Wave
Data Rate	High (cellular 4G/5G)	Very low (250 kbps max)

Application

Industrial telemetry,
smart metersSmart agriculture,
weather stations**Q5. Explain SCADA protocol and its use in industrial IoT.**

SCADA AND BACNET PROTOCOL STANDARDIZATION ARCHITECTURE

Fig: SCADA Architecture Block Diagram):

SCADA (Supervisory Control and Data Acquisition) is an industrial control system architecture that uses remote terminal units and PLCs to monitor and control large-scale real-time industrial processes.

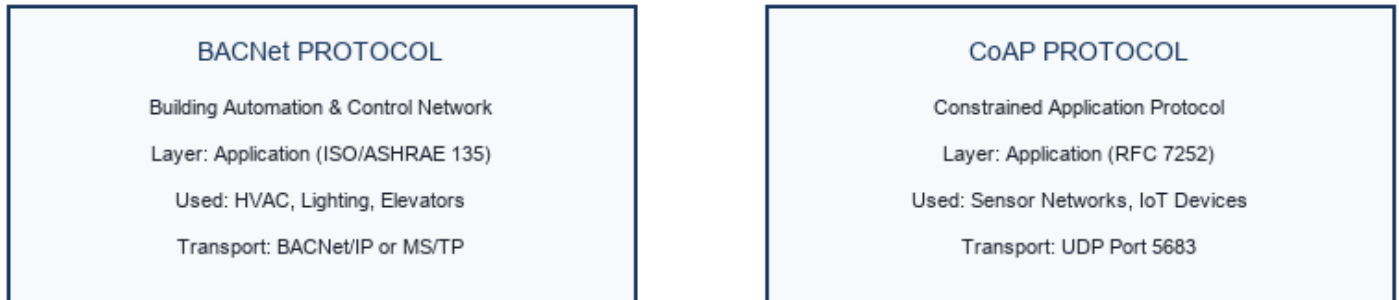
SCADA System Components:

1. Field Sensors & Actuators: Physical sensors measuring pressure, temperature, flow rate.
2. RTU / PLC: Remote Terminal Unit / Programmable Logic Controller — local data acquisition.
3. SCADA Master Station: Central server receiving all RTU data for monitoring.
4. HMI (Human Machine Interface): Graphical operator screen showing live process status.
5. Communication Network: Modbus, DNP3, or IEC 61850 protocols over fiber/cellular.

Use in Industrial IoT:

1. Power Plant Monitoring: Tracks turbine speed, temperature, and electrical output.
2. Water Treatment: Controls water pumps, chlorination, and pipe valve states remotely.
3. Oil & Gas Pipelines: Monitors pressure, flow, and leak detection across 1000s of km.
4. Manufacturing: Automates assembly line conveyor belt speed and robot arm positions.

Q6. Write short notes on: Modbus Protocol / BACNet Protocol



BACNET AND COAP PROTOCOL COMPARISON BLOCK DIAGRAM

Fig: BACNet Protocol Block Diagram

1. Modbus Protocol:

- Oldest and most widely used industrial serial communication protocol.
- Developed by Modicon (now Schneider Electric) in 1979.
- Master-Slave architecture: One master polls multiple slave devices.
- Variants: Modbus RTU (binary, RS-485), Modbus ASCII, Modbus TCP/IP.
- Used in: PLCs, VFDs, power meters, and industrial sensor interfaces.

2. BACNet Protocol (Building Automation and Control Networks):

- ASHRAE standard (ISO 16484-5) for building automation systems.
- Controls HVAC heating/cooling, lighting, elevator, and fire alarm systems.
- Transport: BACNet/IP (over Ethernet) or BACNet MS/TP (over RS-485).
- Used in: Hotels, hospitals, airports, and large commercial buildings.

Q7. What is protocol standardization? Why is it needed in IoT? [Short — 3-5 Marks]

Protocol standardization in IoT is the process of defining common agreed-upon communication rules that all IoT devices and platforms must follow to ensure interoperability.

Why It Is Needed (5 Reasons):

1. Interoperability: Devices from different manufacturers communicate without custom code.
2. Plug & Play: New devices join the network instantly without manual integration.
3. Security: Standardized protocols include built-in encryption and authentication.
4. Reduces Costs: No need to develop custom interfaces for each vendor.
5. Global Scalability: One standard works across millions of devices worldwide.

Q8. Explain network layer functions in IoT protocol stack.

Network Layer Functions in IoT:

1. IP Addressing: Assigns unique IPv6 addresses to identify each IoT device.
2. Packet Routing: Forwards data packets across multi-hop networks to destination.
3. Header Compression: 6LoWPAN compresses 40-byte IPv6 headers to 2-3 bytes.
4. Mesh Routing: RPL protocol discovers best routes through wireless sensor mesh.
5. Fragmentation: Splits large IPv6 packets into small 127-byte 802.15.4 frames.

Q9. List security mechanisms used in IoT protocols. [

Security Mechanisms in IoT Protocols:

1. TLS/DTLS Encryption: Secures data in transit using AES-128/256 ciphers.
2. PKI Certificates: Each device has a unique X.509 digital certificate for identity.
3. MQTT Authentication: Username/password + TLS client certificates.
4. CoAP DTLS: Secures CoAP over UDP using Datagram TLS protocol.
5. Zigbee Link Keys: AES-128 encrypted symmetric keys shared between devices.
6. OTA Update Signing: Firmware updates signed with private key to prevent tampering.



Anjad Ali
PUBLISHER

INTERNET OF THINGS (IoT)

Top 10 'Must-Prepare' Questions & Solutions

(Cross-Unit, Highest Repetition Probability for Exams)

Council for Technical Education and Vocational Training (CTEVT)

Diploma in Computer / Information Technology Engineering

Compiled from VVI Model Questions (Units 1 to 6)

Q1. Explain IoT architecture with a neat diagram .

The Internet of Things (IoT) is a global network of physical devices ('things') embedded with sensors, software, and connectivity to collect and exchange data over the internet without human intervention.

1. Three-Layer IoT Architecture:

DIAGRAM: 3-Layer IoT Architecture Model

APPLICATION LAYER	(Smart Home, Health, Smart Grid)
NETWORK / TRANSPORT	(Wi-Fi, 4G/5G, Zigbee, IPv6, Gateway)
PERCEPTION / SENSING	(Sensors, Actuators, RFID Tags)

2. Five-Layer IoT Architecture:

DIAGRAM: 5-Layer Extended IoT Architecture Model

BUSINESS LAYER	(Executes business models & graphs)
APPLICATION LAYER	(User UI, Smart Cities, Health App)
PROCESSING LAYER	(Cloud, Big Data Analytics, DB)
TRANSPORT LAYER	(Wi-Fi, 5G, MQTT, CoAP, Gateway)
PERCEPTION LAYER	(Sensors, RFID, Cameras, Actuators)

Layer Functions Explained:

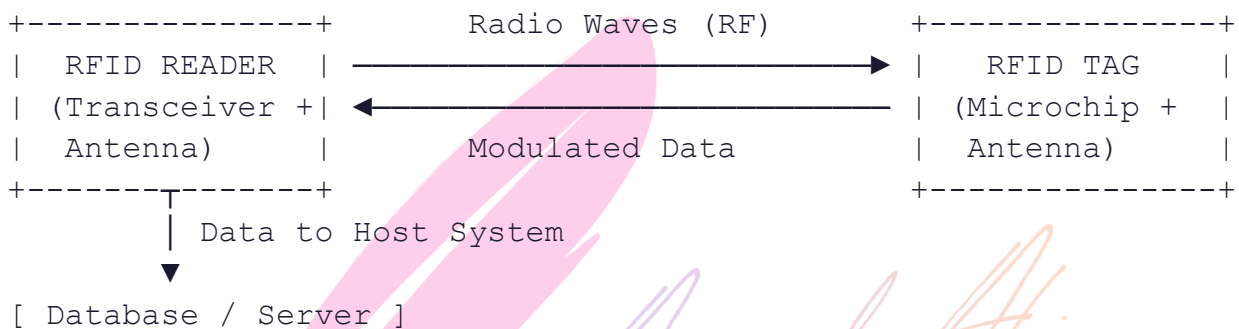
1. Perception Layer: Collects real-world physical data using sensors (temperature, humidity, PIR) and actuators.

2. Transport Layer: Transfers collected sensor data to the processing layer via wireless/wired networks (Wi-Fi, 4G, 6LoWPAN).
3. Processing / Middleware Layer: Stores, analyzes, and processes massive sensor data using Cloud computing and databases.
4. Application Layer: Provides user interfaces and smart services (Smart Homes, Industrial Automation, Smart Health).
5. Business Layer: Manages overall IoT system logic, privacy, business models, and user access.

Q2. Explain RFID technology and its working principle. Differentiate Active vs Passive RFID.

Radio Frequency Identification (RFID) is a wireless technology that uses electromagnetic fields to automatically identify and track tags attached to objects.

DIAGRAM: RFID System Components & Working Principle



Working Principle of RFID:

1. RFID Reader emits radio waves through its antenna.
2. RFID Tag enters the electromagnetic field and picks up the radio signal.
3. Tag uses the RF energy (passive) or internal battery (active) to power its microchip.
4. Tag transmits its stored unique Identification Code (EPC) back to the reader.
5. Reader decodes the signal and sends data to the host computer/cloud for processing.

Comparison: Active RFID vs Passive RFID:

Active RFID Tag	Passive RFID Tag
Internal Battery powered	No battery (Powered by Reader's RF field)
Long range (up to 100 meters+)	Short range (10 cm up to 10 meters)
Expensive (\$10 to \$50 per tag)	Very cheap (\$0.05 to \$0.50 per tag)
3 to 5 years (battery limited)	Unlimited lifespan (no battery to drain)
Large memory (up to 128 KB)	Small memory (up to 1 KB / EPC code)
Vehicle tracking, shipping containers	Retail items, asset tags, library books, passports

Q3. Explain Zigbee architecture and protocol stack. Compare CoAP vs HTTP protocols.

1. Zigbee Architecture & Protocol Stack:

Zigbee is a low-cost, low-power, wireless mesh networking standard designed for short-range control and monitoring applications based on IEEE 802.15.4.

DIAGRAM: Zigbee Protocol Stack Layers.

APPLICATION LAYER	(Application Objects & ZDO Profile)
NETWORK / SECURITY	(Mesh Routing, Security Keys)
MAC LAYER	(IEEE 802.15.4 Medium Access)
PHYSICAL (PHY) LAYER	(IEEE 802.15.4 2.4 GHz RF Radio)

- PHY Layer (IEEE 802.15.4): Operates at 2.4 GHz RF band, providing 250 kbps data rate.
- MAC Layer: Handles channel access (CSMA/CA), frame validation, and beacon management.
- Network (NWK) Layer: Manages mesh network topology, routing, and joining/leaving of nodes.
- Application Layer (APL): Defines device profiles (Smart Energy, Home Automation).

2. CoAP (Constrained Application Protocol) vs HTTP Comparison:

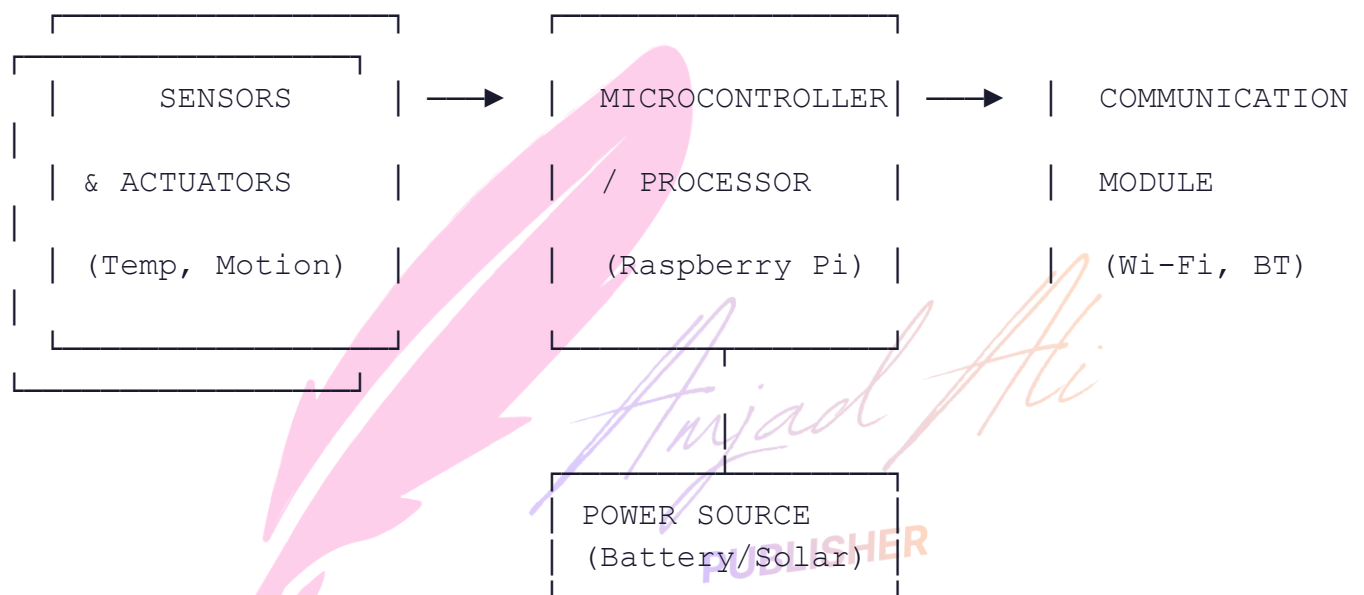
Feature	CoAP (Constrained Application Protocol)	HTTP (Hypertext Transfer Protocol)
Transport Protocol	UDP (User Datagram Protocol)	TCP (Transmission Control Protocol)
Header Size	Fixed 4 Bytes (Extremely lightweight)	Large headers (Hundreds of Bytes)
Target Devices	Constrained IoT microcontrollers (RAM < 10KB)	Resource-rich computers & servers
Resource Discovery	Built-in support (`.well-known/core`)	No native resource discovery mechanism

Communication Model	Asynchronous Publish/Observe & Request/Response	Synchronous Request / Response only
Power Consumption	Ultra-low power consumption	High power consumption

Q4. Explain the building blocks of an IoT device and Raspberry Pi hardware interfaces. [

1. Core Building Blocks of an IoT Device:

DIAGRAM: IoT Device Building Blocks



1. Sensing / Actuation Block: Converts physical parameters (temp, pressure) into electrical signals or performs physical motion.
2. Processing Block: Microcontroller or Single Board Computer (e.g., Raspberry Pi, ESP32) that runs code and processes data.
3. Communication Block: Wireless/wired modules (Wi-Fi, Bluetooth, Zigbee) to connect device to internet.
4. Power Supply Block: Provides electrical power via battery, solar panel, or adapter.

2. Raspberry Pi Hardware Interfaces Explained:

Interface Name	Full Form	Pin Count / Specs	Primary Use in IoT
GPIO	General Purpose Input/Output	40 Pins	Connecting LEDs, switches, relays, digital sensors
I2C	Inter-Integrated Circuit	2 Wires (SDA, SCL)	Connecting multiple sensors (OLED, RTC, MPU6050) on same bus
SPI	Serial Peripheral Interface	4 Wires (MOSI, MISO, SCLK, CS)	High-speed communication with displays and SD cards
UART	Universal Asynch. Receiver-Transponder	2 Wires (TX, RX)	Serial communication with GPS modules, GSM modems

Q5. Explain IoT security requirements and the layered attacker model. [Unit 5 | .

1. Core IoT Security Requirements (CIA Triad + IoT Specifics):

1. Confidentiality: Ensuring sensor data is encrypted so unauthorized attackers cannot read sensitive data.
2. Integrity: Ensuring sensor data and commands are not altered or tampered with during transmission.
3. Availability: Ensuring IoT services and devices remain accessible and operational without DDoS disruptions.
4. Authentication: Verifying the true identity of IoT devices before allowing network access.
5. Non-Repudiation: Ensuring a device or user cannot deny a performed action or sent message.

2. Layered Attacker Model in IoT Security:

DIAGRAM: Layered Attacker Model

Layer 4: Application Attackers → SQL Injection, Web UI Hacking, Data Theft

Layer 3: Network Attackers → Man-in-the-Middle (MitM), Packet Sniffing, DDoS

Layer 2: Firmware Attackers → Malicious Firmware Flashing, Buffer Overflow

Layer 1: Physical Attackers → Hardware Tampering, Stealing Device, Flash Readout

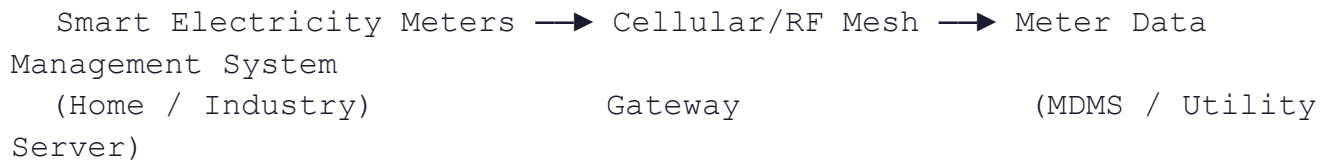
Attack Layer	Attacker Strategy	Common Attack Types	Security Countermeasure
Physical Layer	Direct physical access to device	Side-channel attack, chip extraction	Tamper-proof casing, secure boot
Network Layer	Intercepting network traffic	Eavesdropping, MitM, Replay attack	TLS/DTLS Encryption, WPA3
Software / App Layer	Exploiting code vulnerabilities	Buffer overflow, Unauthorized API access	Secure coding, API authentication

Q6. Explain Smart Grid architecture, AMI, and e-Health Body Area Networks (BAN). [Unit 6 |

1. Smart Grid & Advanced Metering Infrastructure (AMI):

A Smart Grid is an electricity network that uses digital IoT sensors and 2-way communications to monitor and control electricity flow in real-time.

DIAGRAM: AMI Architecture in Smart Grid

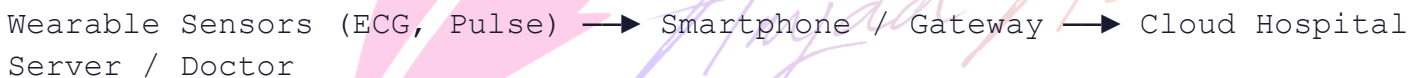


- **Smart Metering:** Replaces manual meter reading with automated, real-time power consumption tracking.
- **Two-Way Communication:** Enables utility companies to remotely connect/disconnect power and adjust electricity rates during peak hours.

2. e-Health Body Area Network (BAN):

A Body Area Network (BAN) consists of wearable or implantable medical IoT sensors (ECG, pulse oximeter, glucose monitor) that continuously monitor a patient's vital signs.

DIAGRAM: e-Health BAN Architecture



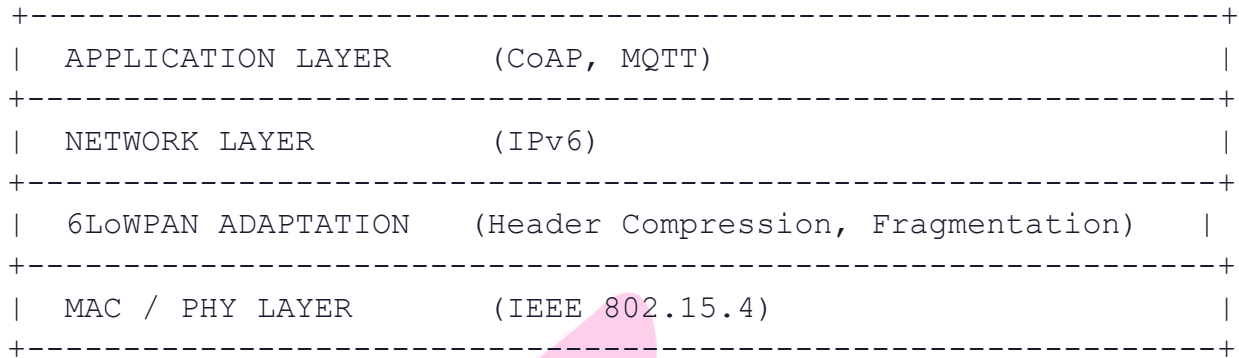
Q7. Differentiate between IoT and M2M (Machine-to-Machine).

M2M (Machine-to-Machine)	IoT (Internet of Things)
Point-to-Point direct communication	Multipoint to Cloud / IP Network
Proprietary or cellular protocols	Open IP protocols (IPv6, MQTT, CoAP)
Cellular / Wired networks	Internet / Cloud IP networks
Limited scalability	Massively scalable (Billions of devices)
Data shared only between connected machines	Data shared across cloud apps & analytics
Hardware dependent (SIM / Modem)	Software & Cloud driven

Q8. Explain 6LoWPAN protocol and its importance in IoT.

6LoWPAN stands for IPv6 over Low-Power Wireless Personal Area Networks. It is an adaptation layer standard that allows IPv6 packets to be transmitted over low-power, constrained IEEE 802.15.4 networks.

DIAGRAM: 6LoWPAN Layer Position



Why 6LoWPAN is Important:

1. Header Compression: Compresses 40-Byte IPv6 headers down to 2-7 Bytes so they fit into small 127-Byte 802.15.4 frames.
2. Fragmentation & Reassembly: Splits large IPv6 packets into smaller fragments for transmission.
3. Direct IP Addressing: Enables tiny battery-powered sensors to have unique IPv6 addresses and connect directly to the internet without complex gateways.

PUBLISHER

Q9. Write a Python program for a basic Raspberry Pi IoT task (Blinking LED / Reading Sensor Data).

This Python program uses the `RPi.GPIO` library to blink an LED connected to GPIO Pin 18 of a Raspberry Pi at 1-second intervals.

PYTHON CODE / SCRIPT: Python Script: Raspberry Pi LED Blink

```
# Import required libraries
import RPi.GPIO as GPIO
import time

# Set GPIO pin numbering mode
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)

# Define GPIO pin for LED
LED_PIN = 18
GPIO.setup(LED_PIN, GPIO.OUT)

print('IoT Task: Starting LED Blink Program...')
try:
    while True:
        GPIO.output(LED_PIN, GPIO.HIGH) # Turn LED ON
        print('LED State: ON')
        time.sleep(1.0)                  # Delay 1 second

        GPIO.output(LED_PIN, GPIO.LOW)  # Turn LED OFF
        print('LED State: OFF')
        time.sleep(1.0)                  # Delay 1 second

except KeyboardInterrupt:
    print('Program stopped by user.')
finally:
    GPIO.cleanup()                       # Reset GPIO pins
    print('GPIO Cleaned Up successfully.')
```

Q10. Discuss cloud storage models, APIs, and AWS IoT services for IoT.

Cloud computing provides elastic storage, processing power, and data analytics capabilities for massive volumes of sensor data generated by IoT devices.

1. IoT Cloud Storage Models:

- Private Cloud: Infrastructure dedicated to a single enterprise (high security, expensive).
- Public Cloud: Shared cloud resources managed by third-party providers (AWS, Azure, Google Cloud).
- Hybrid Cloud: Combines private cloud for sensitive data with public cloud for scalable processing.

2. IoT Cloud APIs:

- REST API: HTTP-based stateless API using GET, POST, PUT, DELETE methods.
- MQTT API: Lightweight publish/subscribe API designed for constrained networks.

3. AWS IoT Core Services:

1. AWS IoT Core: Allows IoT devices to connect securely to AWS cloud via MQTT/HTTP.
2. Device Shadow: A JSON document that stores state information of an IoT device even when offline.
3. AWS IoT Analytics: Processes and analyzes large-scale sensor data.
4. AWS Greengrass: Enables edge computing by running local AWS Lambda code on IoT gateway devices.

PUBLISHER